5-12-2020

# Leveraging Model Flexibility and Deep Structure: Non-Parametric and Deep Models for Computer Vision Processes with Applications to Deep Model Compression

Anthony D. Rhodes
*Portland State University*

# Leveraging Model Flexibility and Deep Structure: Non-Parametric and Deep Models for Computer Vision Processes with Applications to Deep Model Compression

by

Anthony D. Rhodes

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Mathematical Sciences

Dissertation Committee:
Bin Jiang, Chair
Melanie Mitchell, Co-Chair
Mau Nam Nguyen
Wayne Wakeland

Portland State University
2020

**Abstract**

My dissertation presents several new algorithms incorporating non-parametric and deep learning approaches for computer vision and related tasks, including object localization, object tracking and model compression.

With respect to object localization, I introduce a method to perform active localization by modeling spatial and other relationships between objects in a coherent "visual situation" using a set of probability distributions. I further refine this approach with the *Multipole Density Estimation with Importance Clustering* (MIC-Situate) algorithm. Next, I formulate active, "situation" object search as a Bayesian optimization problem using Gaussian Processes. Using my *Gaussian Process Context-Situation Learning* (GP-CL) algorithm, I demonstrate improved efficiency for object localization over baseline procedures. In subsequent work, I expand this research to frame object tracking in video as a temporally-evolving, dynamic Bayesian optimization problem. Here I present the *Siamese-Dynamic Bayesian Tracking Algorithm* (SDBTA), the first integrated dynamic Bayesian optimization framework in combination with deep learning for video tracking. Through experiments, I show improved results for video tracking in comparison with baseline approaches. Finally, I propose a novel data compression algorithm, *Regularized L21 Semi-NonNegative Matrix Factorization* (L21 SNF) which serves as a general purpose, parts-based compression algorithm, applicable to deep model compression.

## Dedication

*This dissertation is dedicated to my father and mother, Jack and Genie Rhodes for their lifetime of support, for which I am eternally grateful; I am especially appreciative of my father for stirring in me an unwavering curiosity and interest in mathematics and science more generally ab initio. This dissertation is additionally dedicated to my wife and life-companion, Shawna, and to my son, Perseus Rhodes.*

**Acknowledgments**

I would like to thank the many professors, teachers, educators, professional colleagues, and not least of all, my own students, who have personally helped and inspired me throughout my academic journey.

Above all, I must thank my co-advisors, Dr. Melanie Mitchell and Dr. Bin Jiang, for their immense support and advocacy throughout this process; I wish to thank my dissertation committee members Dr. Mau Nam Nguynen and Dr. Wayne Wakeland for their time and effort; in addition, I wish to thank Dr. Bruno Jednyak for his mentorship.

## Table of Contents

## List of Tables

## List of Figures

## List of Algorithms

# Chapter 1

# Introduction

The focus of the present work is in non-parametric and deep learning approaches to several of the quintessential problems in computer vision and machine learning at large, including object localization and video tracking. I furthermore advance novel techniques supporting algorithmic efficiency for these tasks, including a method to achieve deep model compression.

Computer vision is a diverse and challenging problem domain, encompassing active research at the intersection of machine learning, artificial intelligence, robotics, general automation, graphics, and medicine – among many other fields. Over the course of the last several years in particular, deep learning and non-parametric methods have emerged as two of the dominant paradigms across not only computer vision, but machine learning more generally. Among their copious attractive attributes, deep learning and non-parametric models offer a representational flexibility and richness that is distinctly amenable to the large data and complexity demands required by many modern machine learning tasks.

Deep learning has, moreover, recently come to occupy a singular role in machine learning as the *de facto* tool for many data-driven problems. While deep learning has achieved extraordinary new benchmarks across a myriad of disciplines, it is nevertheless a field that is still in a state of relative infancy – replete with theoretical lacunae and as-of-yet unexploited synergies with other well-grounded methodologies.

Deep learning is generally best suited for pattern recognition tasks in large data regimes, whereas, conversely, deep models are often deficient in small data domains and prove particularly brittle for problems exhibiting a high degree of situational specificity. Non-parametric methods, by contrast, are generally responsive to small data problems, more robust to situational specificity, and offer the additional benefit of naturally accommodating statistically-principled practices, including Bayesian learning.

The primary goal of my dissertation is to harness and accentuate the benefits of both non-parametric and deep learning approaches for computer vision tasks, and to additionally coherently unify these approaches – when possible. In addition, I provide a matrix factorization algorithm that can be applied to general deep learning models in computer vision for improved efficiency in deep learning tasks. I hypothesize that non-parametric and deep learning approaches can be successfully allied to provide solutions to a wide variety of tasks in computer vision. Furthermore, I posit that this hybrid approach can be leveraged to furnish useful higher order analytic modalities (e.g. "visual situation recognition", search confidence/uncertainty measures) that traditional deep learning approaches alone cannot provide. Indeed, current research has demonstrated, for example, strong evidence that hybrid machine learning methods (*a fortiori*: deep learning combined with Bayesian methods) are requisite for the deployment of deep models in real-world settings for which *epistemic uncertainty* garners increased public risk [139].

To this end, I hypothesize that by exploiting model flexibility and deep structure with non-parametric and deep models, respectively, it is possible to improve upon state-of-the-art methods across a variety of essential computer vision tasks. In total, my dissertation comprises the following six original pieces of research:

1. "Active Object Localization in Visual Situations." arXiv: 1607.00548.

2. “Fast On-Line Kernel Density Estimation for Active Object Localization.” IJCNN, 2017.

3. “Bayesian Optimization for Refining Object Proposals.” IPTA, 2017.

4. “Gaussian Processes with Context-Supported Priors for Active Object Localization.” IJCNN, 2018.

5. “Deep Siamese Networks with Bayesian non-Parametrics for Video Object Tracking.” Future Technologies Confererence, Springer, 2019.

6. “Regularized L21-Based Semi-NonNegative Matrix Factorization with Applications to Deep Model Compression.” (to be submitted for publication)

In (1) and (2), my primary hypothesis is that prior knowledge and situation-relevant context can be efficiently encoded using a dynamic, non-parametric “situation model.” Furthermore, this model, when used in conjunction with deep learning, can accurately and efficiently localize relevant objects in an image – even in the case of partially-observed and sparse data. The experimental successes of (1) and (2) prompted me to consider a more statistically-principled approach to object localization. In (3) and (4), I specifically hypothesize that object localization can be made more efficient still by utilizing deep model-based object detection in combination with non-parametric Bayesian active search. Following this work, I extend this approach to the more extreme, “one-shot” use-case of video tracking. My primary hypothesis in (5) is that video tracking can be framed as a temporally-evolving optimization problem that can be efficiently solved with dynamic Bayesian optimization. In this work I present the first integrated dynamic Bayesian optimization framework in combination with deep learning for video tracking. In (6) I posit that exploiting a particular loss function (L21) in combination with semi-nonnegative constraints, can engender an

effective general data compression algorithm that is particularly well-suited to highly overdetermined systems, including deep models.

# Chapter 2

# Active Object Localization in Visual Situations

## 2.1 Overview

In this chapter, I collaborated with Quinn et al. [76] in developing a method to perform active localization of objects in instances of visual situations. The system, called "Situate," combines given and learned knowledge of the structure of a particular situation, and adapts that knowledge to a new situation instance as it actively searches for objects. The general research question taken up through this project was how to effectively encode situation-relevant context applied to a dynamic, active search of a "visual situation."

More concretely, the system learns a set of probability distributions describing spatial and other relationships among relevant objects. These distributions are then used to iteratively sample object proposals on a test image; concurrently, the system uses information from those object proposals to adaptively modify the distributions depending on what the system has detected. For this research, I contributed to designing and integrating Situate's active object localization models.

## 2.2 Background and Related Work

Most object-localization algorithms in computer vision do not exploit prior knowledge or dynamic perception of context. The current state-of-the-art methods employ feed-

forward deep networks that produce and test a fixed number of object proposals (also called region proposals)—on the order of hundreds to thousands—in a given image (e.g., [19]– [21]). An object proposal is a region or bounding box in the image. Assuming an object proposal defines a bounding box, the proposal is said to be a successful localization (or detection) if the bounding box sufficiently overlaps a target object's ground-truth bounding box.

In fact, state-of-the-art methods are still susceptible to several problems, including difficulty with cluttered images, small or obscured objects, and inevitable false positives resulting from large numbers of object-proposal classifications. Moreover, such methods require large training sets for learning, and potential scaling issues as the number of possible categories increases. For these reasons, several groups have pursued the more human-like approach of "active object localization," in which a search for objects unfolds over time, with each subsequent time step using information gained in previous time steps (e.g., [24]–[26]).

Work on active object localization has a long history in computer vision, often in the context of active perception in robots [182] and modeling visual attention [184]. The literature of this field is large and currently growing — here I summarize a few examples of recent work most similar to the present work. Alexe et al. [183] propose an active, context-driven localization method: given an initial object proposal in a test image, at each time step the system uses a nearest-neighbor algorithm to find training image regions that similar in position and appearance to the current object proposal. These nearby training regions then "vote" on the next location to propose in the test image, given each training region's displacement vector relative to the ground-truth target object. The systems outputs the highest scoring of all the visited windows, using an object-classifier score. While this method is generally effective, the nearest-neighbor method can be computationally expensive. A more

efficient and accurate version of this method, using R-CNN object proposals and random forest classifiers is described in [185]. Other groups have used recurrent neural networks (RNNs) to perform active localization, see: Mnih et al. [186]. Still, other researchers frame active object localization as a Markov decision process (MDP) and use reinforcement learning to learn a search policy. The approach proposed in [187] involves learning a search policy for a target object that consists of a sequence of bounding-box transformations. In the MDP method proposed in [188], an action consists of running a detector for a "context class" that is meant to help locate instances of the target "query class". Nagaraja et al. [189] propose an MDP method in which a search policy is learned via "imitation learning": in a given state, an oracle demonstrates the optimal action to take and the policy subsequently learns to imitate the oracle. Like these methods, my approach focuses on perception as a temporal process in which information is used as it is gained to narrow the search for objects.

The key differences between the current approach and these other active localization methods can be summarized as follows: often these methods are tested on datasets in which the role of context is limited; these methods often rely on exhaustive co-occurrence statistics among object categories; and it is usually hard to understand why these methods work well or fail on particular object categories. In addition, the reinforcement learning based methods learn a policy that is fixed at test time; in the current method, the representation of a situation itself adapts (via modifications to probability distributions) as information is obtained by the system. Finally, the amount of training data and computation time required can be quite high, especially for reinforcement learning and RNN-based methods.

## 2.3 Methodological Summary

My approach is an example of active object localization, but in the context of specific situation recognition. Thus, only objects specifically relevant to the given situation are required to be located. Situate is provided some prior knowledge— the set of the relevant object categories—and it learns (from training data) a representation of the expected spatial and semantic structure of the situation. This representation consists of a set of joint probability distributions linking aspects of the relevant objects. Then, when searching for the target objects in a new test image, the system samples object proposals from these distributions, conditioned on what has been discovered in previous steps. That is, during a search for relevant objects, evidence gathered during the search continually serves as context that influences the future direction of the search. My hypothesis is that this localization method, by combining prior knowledge with learned situation structure and active context-directed search, will require dramatically fewer object proposals than methods that do not use such information.

For the purposes of this particular research, I focus on a relative simple visual "situation", that of "dog walking", where each image contains exactly one (human) dog-walker, one dog, one leash, and unlabeled "clutter" (such as non-dog-walking people, buildings, etc). There are 500 such images in my dataset. My system's task is to locate three objects—Dog-Walker, Dog, and Leash—in a test image using as few object proposals as possible.

For brevity's sake, I will now describe only some of the details of the prior and "situation model" of my system. For each of the three object categories, Situate takes the ground-truth bounding boxes from the training set, and fits the natural logarithms of the box sizes (area ratio) and box shapes (aspect ratio) to normal distributions. At test time, the system uses these prior distributions to sample area ratio and aspect

ratio.

From training data, Situate learns a situation model: a set of joint probability distributions that capture the "situational" correlations among relevant objects with respect to location, area, and aspect ratio. When running on a test image, Situate will use these distributions in order to compute category-specific location, area, and aspect ratio probabilities conditioned on objects that have been detected.

To model the locations of objects, I take a vector $\theta_{xy}$ from each training image where $\theta_{xy} = (x_{dog}, y_{dog}, x_{DW}, y_{DW}, x_{leash}, y_{leash})$, where DW indicates the dog walker. I then model these vectors with a normal distribution $N(\mu_{\theta_{xy}}, \Sigma_{\theta_{xy}})$. Although the true distribution may not be normal, a simple normal distribution helps to encapsulate that these variables are indeed related to one another, and that Situate should update its search priorities if any of these variables are known.

Once an object has entered the Workspace, parameters from Workspace objects and the joint distribution are used to generate a conditional distribution. I can then sample new bounding box locations for a particular object type such that

$$(x_{sample}, y_{sample}) \sim N(\mu_{\theta_{xy}}, \Sigma_{\theta_{xy}} | (x_{w_1}, y_{w_1}, ..., x_{w_n}, y_{w_n})) \tag{2.1}$$

where $x_{w_i}$ and $y_{w_i}$ represent the $x$ and $y$ coordinates of $i^{th}$ object in the Workspace. The location does not fully define a bounding box. I also need to generate a bounding box shape and size. I model the shape and size using the log aspect ratio of the box $\gamma$ (i.e., the log of the ratio of box width to box height) and the log of the area ratio $\delta$ (i.e., the log of the ratio of the box area to the image area). These values are gathered from training images for each object type and are modeled such that

$$\gamma_{object} \sim N(\mu_{\gamma_{object}}, \sigma_{\gamma_{object}}) \tag{2.2}$$

and

$$\delta_{object} \sim N(\mu_{\gamma_{object}}, \sigma_{\delta_{object}}) \tag{2.3}$$

Once an object has entered the Workspace, I use a conditional distribution to generate bounding box sizes and shapes. To build the conditional distribution, I first model the joint distribution using the training data such that:

$$\theta_{\gamma\delta} \sim N(\mu_{\Theta_{\gamma\delta}}, \Sigma_{\Theta_{\gamma\delta}}) \tag{2.4}$$

where

$$\Theta_{\gamma\delta} = (\gamma_{dog}, \delta_{dog}, \gamma_{DW}, \delta_{DW}, \gamma_{leash}, \delta_{leash}). \tag{2.5}$$

I then sample a new box shape and size such that

$$(\gamma_{object}, \delta_{object}) \sim N(\mu_{\Theta_{\gamma\delta}}, \Sigma_{\theta_{\gamma\delta}} | (\gamma_{w_1}, \delta_{w_1}, ..., \gamma_{w_n}, \delta_{w_n})) \tag{2.6}$$

where $\gamma_{w_i}$ and $\delta_{w_i}$ are the $\gamma$ and $\delta$ values from the $i^{th}$ entry in the Workspace. For (2.6) I use the mutivariate conditioning formula:

$$N(\boldsymbol{x_1}|\boldsymbol{x_2} = \boldsymbol{a}) \sim N(\boldsymbol{\mu_1} + \Sigma_{12}\Sigma_{22}^{-1}(\boldsymbol{a} - \boldsymbol{\mu_2}), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}) \tag{2.7}$$

Once I have $(x_{sample}, y_{sample}, \gamma_{sample}, \delta_{sample})$, a sampled bounding box is fully specified.

In Figures 2.1-2.3 below, I illustrate an example run of the Situate pipeline for active object localization.

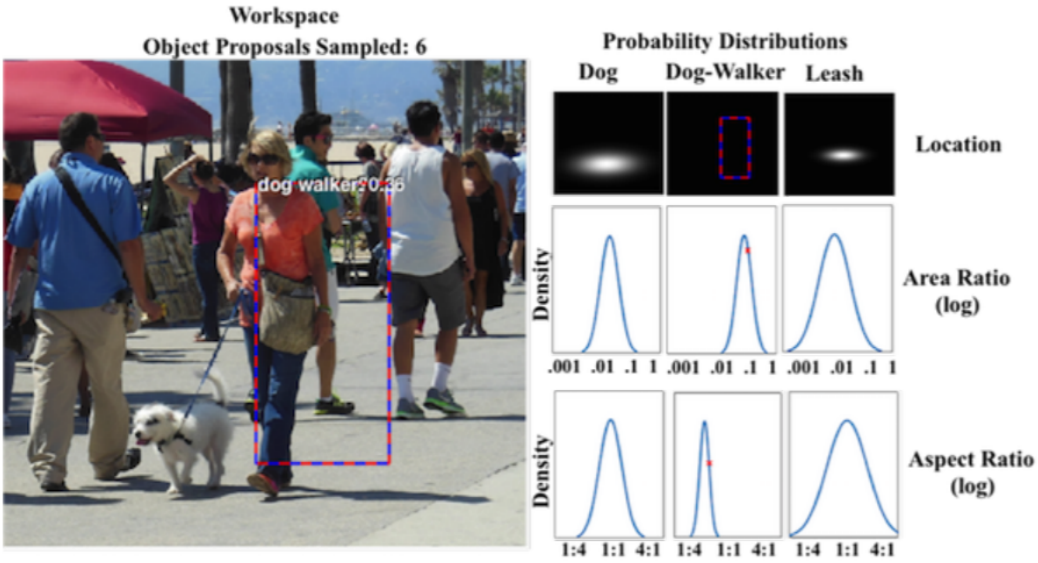


Figure 2.1: The system's state after six object proposals have been sampled and scored. The sixth proposal was for the Dog-Walker category and its score (0.36) is higher than the provisional threshold, so a provisional detection was added to the Workspace (red dashed box; the samples that gave rise to this proposal are shown in red on the various Dog-Walker probability distributions). This causes the location, area ratio, and aspect ratio distributions for Dog and Leash to be conditioned on the provisional Dog-Walker detection, based on the learned situation model. In the location distributions, white areas denote higher probability. The area-ratio and aspect-ratio distributions for Dog and Leash have also been modified from the initial ones, though the changes are not obvious due to the simple visualization.

## 2.4 Experimental Results

The stated hypothesis is that by using prior (given and learned) knowledge of the structure of a situation and by employing an active context-directed search, Situate will require dramatically fewer object proposals to locate relevant objects than methods that do not use such information. In order to test this hypothesis, this work compares Situate's performance with that of four baseline methods and two variations on Situate. The four baseline methods include: uniform sampling, sampling from learned area-ratio and aspect-ratio distributions, using a location prior determined by

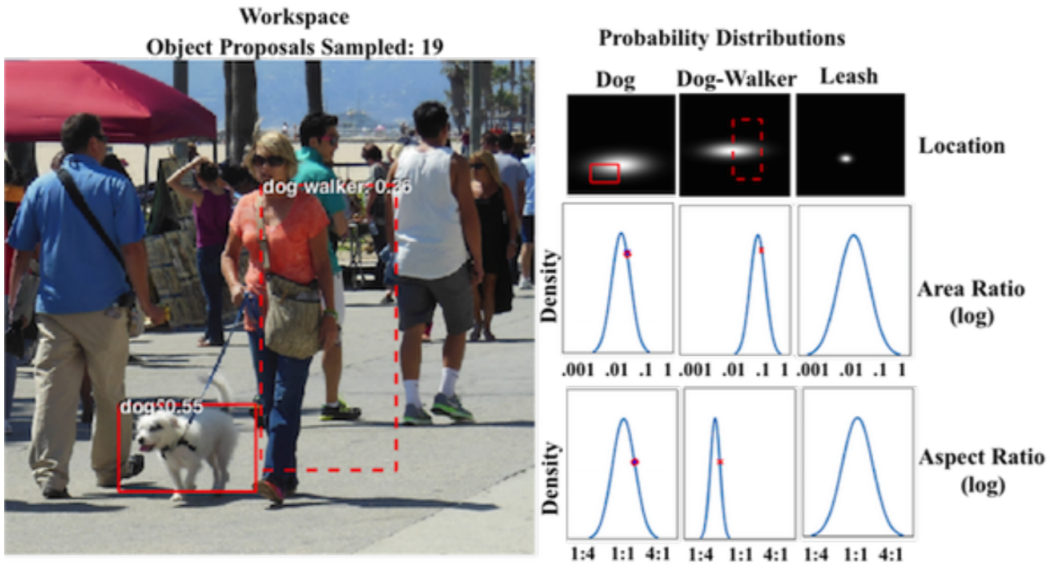


Figure 2.2: The system's state after 19 object proposals have been sampled and scored. The focused conditional distributions have led to a Dog detection at IOU 0.55 (solid red box, indicating final detection), which in turn modifies the distributions for Dog-Walker and Leash. The situation model is now conditioned on the two detections in the Workspace. Note how strongly the Dog and Dog-Walker detections constrain the Leash location distributions.

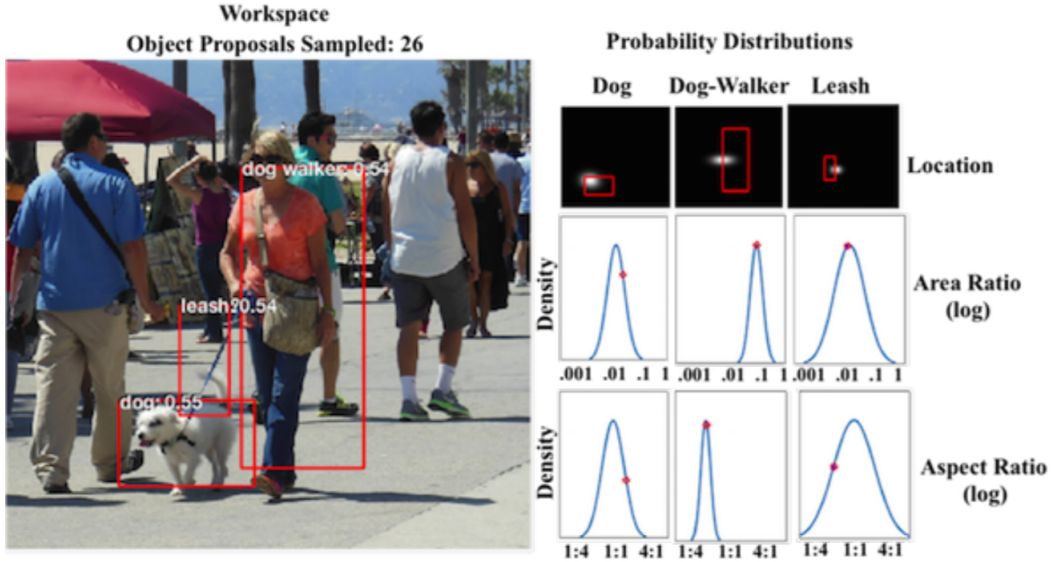


Figure 2.3: The system's state after 26 object proposals have been sampled and scored. Final detections have been made for all three objects.

"salience" calculations, and Randomized Prim's Algorithm, a category-independent object-proposal method.

In reporting results, I use the term *completed situation detection* to refer to a run on an image for which a method successfully located all three relevant objects within 1,000 iterations; I use the term *failed situation detection* to refer to a run on an image that did not result in a completed situation detection within 1,000 iterations.

The various methods described above are characterized by: (1) **Location Prior:** whether the prior distribution on location is uniform or based on salience; (2) **Box Prior:** whether the prior distributions on bounding-box size and shape are uniform or learned; and (3) **Situation Model:** whether, once one or more object detections are added to the Workspace, a learned situation model conditioned on those detections is used instead of the prior distributions, and whether that conditioned model is combined with a salience prior for location.

As described above, the dataset contains 500 images. For each method, I performed 10-fold cross-validation: at each fold, 450 images were used for training and 50 images for testing. Each fold used a different set of 50 test images. For each method I ran the algorithm described previously on the test images, with *final-detection-threshold* set to 0.5, *provisional-detection-threshold* set to 0.25, and *maximum number of iterations* set to 1,000. In reporting the results, I combine results on the 50 test images from each of the 10 folds and report statistics over the total set of 500 test images.

## 2.5 Discussion

The results presented in this chapter strongly support my hypothesis that by using knowledge of a situation's structure in an active search, my method will require dramatically fewer object proposals than methods that do not use such information.

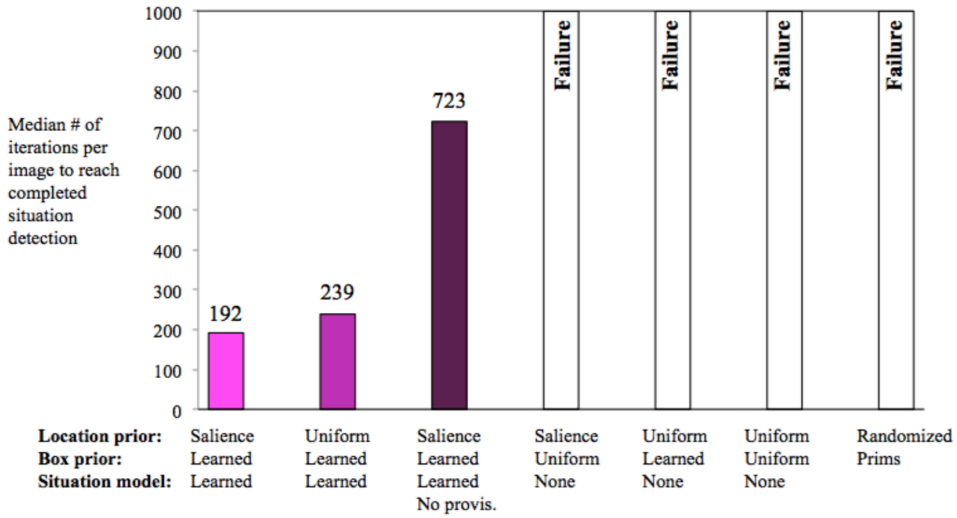


Figure 2.4: Results from seven different methods, giving median number of iterations per image to reach a completed situation detection (i.e., all three objects are detected at final detected threshold). If a method failed to reach a completed situation detection within the maximum iterations on a majority of test images, its median is given as "Failure".

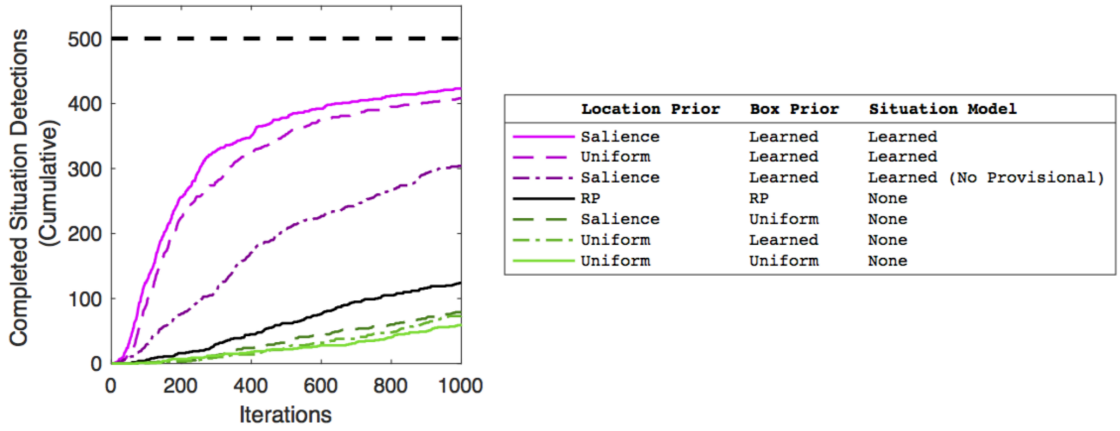


Figure 2.5: Cumulative number of completed situation detections as a function of iterations. For each value n on the x-axis, the corresponding y-axis value gives the number of test image runs reaching completed situation detections with n or fewer object proposal evaluations. "RP" refers to the Randomized Prim's algorithm.

Situate's active search is directed by a set of probability models that are continually updated based on information gained by the system as it searches. My results show that using information from provisional, incomplete detections is key to closing in

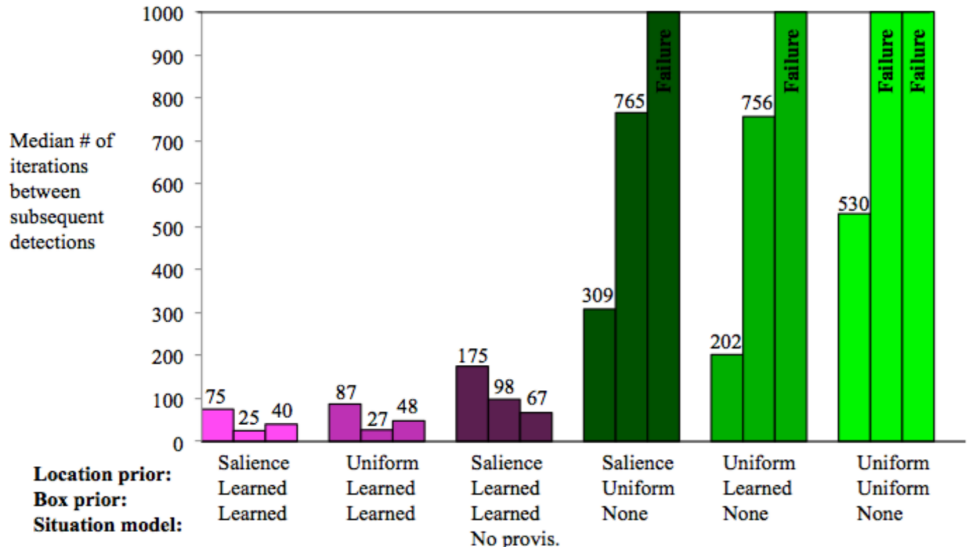


Figure 2.6: Results from different methods giving median number of iterations between subsequent object detections (at the final-detection threshold. For each method, the plot gives three bars: the first bar is $\tilde{t}_{0,1}$, the median number of iterations to the first object detection in that image; the second bar is $\tilde{t}_{1,2}$, the median number of iterations from the first to the second detection; and the third bar is $\tilde{t}_{2,3}$, the median number of iterations from the second to the third detection. Results for Randomized Prim's [181] is not included here because I did not collect this finer-grained data for that method.

quickly on a complete situation detection.

My results showed that a location prior based on a fast-to-compute salience map only marginally improved the speed of localization. More sophisticated salience methods might reduce the number of iterations needed, but the computational expense of those methods themselves might offset the benefits. In general, this work provides the following contributions: (1) It proposes a new approach to actively localizing objects in visual situations, based on prior knowledge, adaptable probabilistic models, and information from provisional detections. (2) Through experiments, I demonstrate the benefits of this approach in addition to analyzing where and why this approach fails. (3) This research contrasts the approach embodied by Situate with several other research groups working on incorporating context into object detection, and on active object localization.

# Chapter 3

# Fast On-Line Kernel Density Estimation for Active Object Localization

## 3.1 Overview

In this chapter, I propose a novel method for prior learning and active object localization of knowledge-driven search in static images using non-parametric kernel density estimation as the basis for a *situation model*. In this system, prior situation knowledge is captured by a set of flexible, kernel-based density estimations that represent the expected spatial structure of the given situation. These estimations are efficiently updated by information gained as the system searches for relevant objects, allowing the system to use context as it is discovered to narrow the search. More specifically, at any given time in a run on a test image, this system uses image features plus contextual information it has discovered to identify a small subset of training images— an *importance cluster*—that is deemed most similar to the given test image, given the context. This subset is used to generate an updated situation model in an on-line fashion, using an efficient *multipole expansion* technique. As a proof of concept, I apply my algorithm to a highly varied and challenging dataset consisting of instances of a "dog-walking" situation. These results support the hypothesis that dynamically-rendered, context-based probability models can support efficient object localization in visual situations. In this work I present an efficient algorithm for computing non-parametric probability density estimates. Unlike parametric methods,

non-parametric methods make no global *a priori* assumptions about the shape of a distribution function. These models are consequently highly flexible and capable of representing useful patterns in diverse datasets.

### 3.1.1 Kernel Density Estimation

Kernel density estimation (KDE) is a widely used method for computing non-parametric probability density estimates from data. Suppose the data lives in a $d$ dimensional space and a set $S$ of training examples $\mathbf{x}$, with $\mathbf{x} \in \mathbb{R}^{\mathbf{d}}$, is given. Now I want to compute the probability density of an unobserved point $\mathbf{z} \in \mathbb{R}^{\mathbf{d}}$, given $S$. The idea of KDE is to use a *kernel function*, which measures similarity between data points, so that points in $S$ that are most similar to $\mathbf{z}$ contribute the most weight to the density estimate at point $\mathbf{z}$.

This concept is formalized as follows. Using a kernel function $K$ and bandwidth parameter $\sigma$, I estimate the density $f$ at a point $\mathbf{z} \in \mathbb{R}^{\mathbf{d}}$ due to $N$ local points, $\mathbf{x_1}, ..., \mathbf{x_N}$ with the following formula:

$$\hat{f}(\mathbf{z}) = \frac{\mathbf{1}}{\sigma^{\mathbf{d}}\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{K}(\mathbf{z} - \mathbf{x_i}) \textbf{ with } \int \mathbf{K}(\mathbf{z})\mathbf{dz} = \mathbf{1} \tag{3.1}$$

My hypothesis is that prior and joint non-parametric distributions will be able to capture likely bounding box widths and heights more flexibly than multivariate Gaussian distributions for these values. To simplify my focus, I retain the original uniform and multivariate Gaussian distributions for the prior location and joint locations models, respectively.

A commonly used kernel function is the Gaussian kernel:

$$K(\mathbf{u}) = (\mathbf{2}\pi)^{-\frac{\mathbf{d}}{\mathbf{2}}} \mathbf{exp}\left\{-\frac{\|\mathbf{u}\|^{\mathbf{2}}}{\mathbf{2}\sigma^{\mathbf{2}}}\right\} \tag{3.2}$$

Which yields the following form for the kernel density estimate of $f$, due to $N$ points:

$$\hat{f}(\mathbf{z}) = \mathbf{Z}\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{exp}\left\{-\frac{\|\mathbf{z}-\mathbf{x_i}\|^{\mathbf{2}}}{\mathbf{2}\sigma^{\mathbf{2}}}\right\} \textbf{ with } \mathbf{Z} = \frac{\mathbf{1}}{\sigma^{\mathbf{d}}\mathbf{N}}(\mathbf{2}\pi)^{-\frac{\mathbf{d}}{\mathbf{2}}} \tag{3.3}$$

I can now express the conditional density estimate for a point $\mathbf{z}$, given observed data $\{\mathbf{y}\}$ and kernel $K$, as follows:

$$\hat{f}(\mathbf{z}|\mathbf{y}) = \frac{\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{K}(\mathbf{z}-\mathbf{x_i^z})\mathbf{K}(\mathbf{y}-\mathbf{x_i^y})}{\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{K}(\mathbf{y}-\mathbf{x_i^y})} \tag{3.4}$$

The complexity associated with computing a density estimate $\hat{f}$ at $M$ discrete values of $\mathbf{z}$, each time using $N$ neighboring points is $O(M \cdot N)$, which is frequently prohibitive for on-line density approximations with large images and/or large values of $N$. Thus, in order to efficiently employ non-parametric models for active object localization, I needed to solve two related problems. First, the system needs to choose—from the training data—a small number $N$ of points that gives the most useful information for the kernel density estimate. Second, even with a small $N$, it can still be expensive to compute this estimate, due to the multiplicative $O(M \cdot N)$ complexity, so I need to construct an accurate and fast density approximation method that scales well with the number of variables on which I will be conditioning the distributions. Towards these ends I develop: (1) a novel method to use the context of the detections discovered so far in the Situate Workspace to determine an importance cluster—an appropriate, information-rich subset of the training data to use to create conditional distributions; and (2) a fast approximation technique for estimating distributions based on the method of multipole expansions.

### 3.2 Context-Based Importance Clustering

My first innovation addresses the problem of determining an appropriate subset of the data to use to compute conditional distributions. Because a dataset of images depicting a particular, sometimes complex, visual situation which is likely to exhibit high variability, I would like to optimally leverage contextual cues as my algorithm discovers them, in order to assist in object localization. As such, I employ a novel *context-based importance clustering* (CBIC) procedure, which my system uses during its active search for objects.

The current procedure instead computes a flexible non-parametric conditional estimate, not from the entire training set, but from a subset of the training images—those that are deemed to be most similar to the test image, given the object proposals currently in the Workspace.

The motivation for this method is that I wish to focus my density estimation procedure on data that is most contextually relevant to a given test image, as it is perceived at a given time in a run.

More specifically, during a run of Situate on a test image, whenever a new object proposal has been added to the Workspace (i.e., the proposal's score is above one of the detection thresholds), I determine a subset of the training data to use to update conditional distributions for the other object categories. To do this, I cluster the training dataset, using a k-means algorithm, based on the following features. (1) In the case where a single object has been localized, I cluster based on the normalized size of that object category's ground-truth bounding boxes. (2) When multiple objects have been localized, I again use the normalized sizes of the located object-categories, but I also use the normalized distance between the localized objects.

One reason for using these particular features is that they are strongly associated

with both the depth of an object in an image as well as the spatial configurations of objects in a visual situation. Together, these data provide the system with useful information about the size of the bounding-box of a target object. The number of clusters used for k-means is rendered optimally from a range of possible values, according to a conventional internal clustering validation measure based on a variance ratio criterion (Calinski-Harabasz index). Once the training data has been clustered, the test image is then assigned to a particular cluster—the *importance cluster*— with the nearest centroid. Note that importance clusters change dynamically as Situate adds new proposals to the Workspace.

### 3.3 Kernel Density Estimation with Multipole Expansions

My second innovation is to employ a fast approximation technique for estimating distributions: the method of multipole expansions. Multipole expansions are a physics inspired method for estimating probability densities with Taylor expansions.

Let $K$ denote the Gaussian kernel. I apply the multipole method to approximate the kernel density estimate due to $N$ points by forming the multivariate Taylor series for $K(\mathbf{z} - \mathbf{x_i})$. The key advantage of this method is that, following the scheme of the factorized Gaussians presented, the kernel estimate about the centroid $\mathbf{x}*$ (i.e., the center of the Taylor series expansion) can be expressed in factored form (I omit the details here for brevity). The multipole form of this factorization is given by the following expression:

$$\sum_{i=1}^{N} K(\mathbf{z} - \mathbf{x_i}) = \mathbf{G}(\mathbf{z}) \odot \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{w_i F}(\mathbf{x_i}) \tag{3.5}$$

Here, the symbol $\odot$ connotes the multiplication of two Taylor series with vector components; $G(\mathbf{z})$ is the Taylor series representing the points $\mathbf{z}$ at which I am estimating

densities, and $F(\mathbf{x_i})$ is the Taylor series representing the elements of the importance cluster being used to estimate these densities. The value $w_i$ weights the point $\mathbf{x_i}$ by how similar it is to the test image, using the features described in the previous subsection.

Note that the sum over the weighted $F$ terms needs to be performed only once in order to estimate $M$ point-wise densities.

Now, suppose one wish to compute a density estimate $\hat{f}$ at $M$ discrete values of $\mathbf{z}$, each time using $N$ neighboring points. As mentioned previously, doing this directly with KDE is $O(M \cdot N)$ complexity. What the multipole method allows is a reasonable approximation to KDE, but with $O(M + N)$ complexity, where $N$ is the size of the importance cluster. This is potentially a huge gain in efficiency; in fact, it allows us to use this method in an on-line fashion while the system performs its active search. In order to use the multipole method in the Situate architecture, I need to extend the previous formulation of $\hat{f}(\mathbf{z}|\mathbf{y})$ to approximate conditional probability densities (e.g., the expected distribution of "dog" widths / heights given a detected "dog-walker").

I can subsequently apply the multipole expansion method to obtain an expression for conditional density estimation with multipole expansion:

$$\hat{f}(\mathbf{z}|\mathbf{y}) \propto \mathbf{G}(\mathbf{K}(\mathbf{z} - \mathbf{x_i}*)) \odot \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{w_i}\mathbf{F}(\mathbf{x_i}) \tag{3.6}$$

In this equation, $\mathbf{x}^*$ is a stochastically determined centroid for the estimate (as will be explained in the next subsection); $G(\mathbf{z})$, $F(\mathbf{x_i})$, and $w_i$ are all defined as before. This expression yields a complexity requirement of O(M + N). My *stochastic filtering* technique obviates the need for expensive pre-clustering techniques that are often applied in kernel density estimation. My proposed stochastic filtering method produces a sparse density. The sparsity of the estimate is the penalty paid for using

a stochastic filter. Nevertheless, so long as $M \gg N$ (a very natural assumption for most practical applications of density estimation), then $\hat{f} \to f(\mathbf{z})$ as $M \to \infty$ which follows from the convergence of the Taylor series.

### 3.4 Stochastic Filtering

A significant issue arises when I consider performing this density approximation for a large $M$ (i.e., for many different point-wise approximations), which might be required in cases for which comprehensive, interpretable models are desired. The issue is that the inevitable errors in the approximation can accumulate.

Although the overall error in the density approximation can be improved by choosing a sufficiently large order for the Taylor expansions (such as a multivariate quadratic, cubic, etc.), the error margin can nonetheless potentially become excessive when aggregated over points that are a great distance from the center of each Gaussian kernel; naturally, this issue is compounded further as the size of the set of sample points, $N$, grows.

There have been a few proposed remedies in the literature to this issue of aggregated errors. The authors in [79] simply suggest limiting the points over which the density estimation is performed to a small subset of the space, but this is a fairly weak and impractical compromise for a general problem setting. Alternatively, the authors in [80] suggest performing a constrained clustering of the density space and then estimating each point-wise density by its nearest centroid. However, finding an appropriate clustering needed for this scheme turns out to be very expensive to achieve. Various approximate solutions exist, including an adaptive, greedy algorithm called "farthest point clustering" [84] and a more computationally-efficient version given by [85].

I introduce a new approach, termed *stochastic filtering*—that obviates the need

for such clustering of the density space. For each target point-density approximation $\hat{f}(z)$, I simply choose one element of the current importance cluster at random, and use this element to be the center of the Taylor expansion $G(z)$.

Note that my proposed stochastic filtering method will produce a *sparse density estimate* since the stochastic choice of cluster center coupled with the Gaussian kernel will render many of the approximate values zero. The sparsity of the estimate is therefore the penalty I pay for using this filter. Nevertheless, so long as $M >> N$ (a very natural assumption for most practical applications of density estimation), then $\hat{f} \rightarrow f(z)$ as $M \rightarrow \infty$, which follows from the convergence of the Taylor series. From a sparse estimate, one can additionally apply a simple Gaussian smoothing process to achieve a low-cost, yet high-fidelity density estimate. It should also be noted that perfect density estimation is not at all required for practical use in our object localization task. Instead I desire an efficient localization process which is capable of dynamically leveraging visual-contextual cues for active object localization.

### 3.5 MIC-Situate Algorithm

The following are the steps in my algorithm, Multipole Density Estimation with Importance Clustering (MIC-Situate). Assume that I have a training set $S$, and Situate is running on a test image $T$. Situate chooses an object category at random, samples a location and a bounding-box width and height from its current distributions for the given object category in order to form an object proposal, and scores that object proposal to determine if it should be added to the Workspace. Suppose that $L$ object proposals have been added to the Workspace, with values $\{l_1, ..., l_L\}$ (e.g.,$l_1$ might be the (width, height) values of a detected dog-walker bounding box, and $l_2$ might be the (width, height) values of a detected dog bounding box.) Whenever a new object proposal is added to the Workspace, do the following: For each object

category $c$:

1. Perform k-means clustering of the training data.
2. Determine which cluster the test image belongs to (the importance cluster).
3. Using this importance cluster, compute the fast multipole conditional density estimation, conditioning on the L detected objects.
4. Update the size (width/height) distribution for object category $c$.

### 3.6 Experimental Results

I generated results from running the methods described above for the MIC-Situate algorithm which utilizes both the novel importance clustering technique as well as the fast non-parametric, multipole method for learning a flexible knowledge representation of bounding-box sizes of objects for active object localization. Altogether, I tested four distinct methods for object localization in the dog-walking situations: (1) **Multipole** (**with IC**): non-parametric multipole method with importance clustering, as described above (2) **Multipole** (**no IC**): non-parametric multipole method without importance clustering (where density approximations are generated using the entire training dataset), (3) **MVN**: distributions learned as multivariate Gaussian methods and (4) **Uniform**: a baseline uniform distribution. In the case of (1) and (2), I used a multipole-based non-parametric density estimate for target object width/height priors, utilizing the entire training dataset; I similarly used conditional multipole density estimates for the conditional width/height size distributions.

This work has provided the following contributions: (1) I propose a new approach to actively localizing objects in visual situations using a knowledge-driven search with adaptable probabilistic models. (2) I devise a general-purpose procedure that

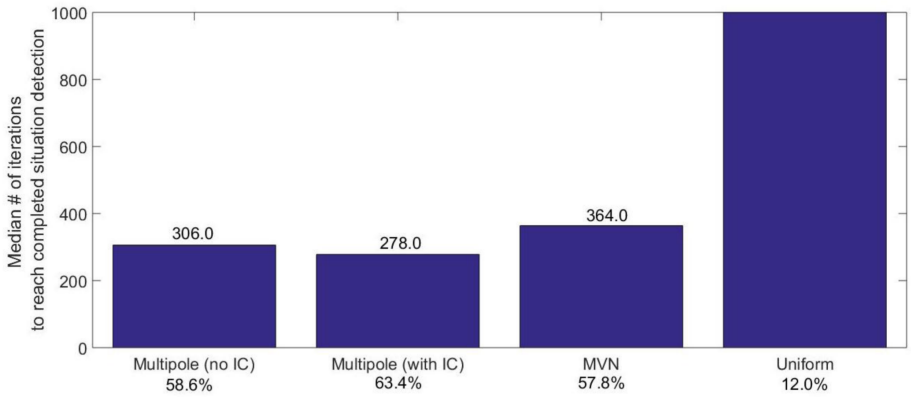


Figure 3.1: Results for the four methods I experimented with for object localization in the Dog-Walking situation images. The graph reports the median number of iterations required to reach a completed situation detection (i.e. correct final bounding-boxes for all three objects). Note that the median value for "Uniform" was "failure"—that is, greater than 1,000. The percentages listed below each graph indicate the percentage of images in the test set for which the method reached a completed situation. For example, the "Multipole (no IC)" method reached completed situations on 58.6% of the 500 images.

uses observed/contextual data to generate a refined, information-rich training set (an importance cluster) applicable to problems with high situational specificity. (3) I develop a novel, fast kernel density estimation procedure capable of producing flexible models efficiently, in a challenging on-line setting; furthermore, when applied in conjunction with importance clustering, this estimation procedure scales well with even a large number of observed variables. (4) I employ these techniques to the problem of conditional density estimation. (5) As a proof of concept, I apply my algorithm to a highly varied and challenging dataset.

# Chapter 4

# Bayesian Optimization for Refining Object Proposals & Gaussian Processes with Context-Supported Priors for Active Object Localization

## 4.1 Overview

Precise object localization remains an enduring, open challenge in computer vision. For example, fine-grained pedestrian localization in images is an active area of research with rich application potential [174]. More generally, accurate object localization is a vital task for many real-word applications of computer vision including: autonomous driving [175], cancer detection [176], image captioning [177], scene recognition [178] and robotics [179]. Current benchmark approaches [180] in object localization commonly apply a form of semi-exhaustive search, requiring a high volume—oftentimes thousands—of potentially expensive function evaluations, such as classifications by a convolutional neural network (CNN). Because of their black box nature, these methods often lack interpretability and neglect to incorporate top-down information including contextual and scene attributes.

With [20][21], Girshick et al. achieved state-of-the-art performance on several object detection benchmarks using a "regions with convolutional neural networks" (R-CNN) approach. R-CNN comprises two phases: the region proposal generation and the proposal classification. Regional proposal generation renders rectangular regions of interest (ROIs) that are later classified by a deep CNN during proposal classifica-

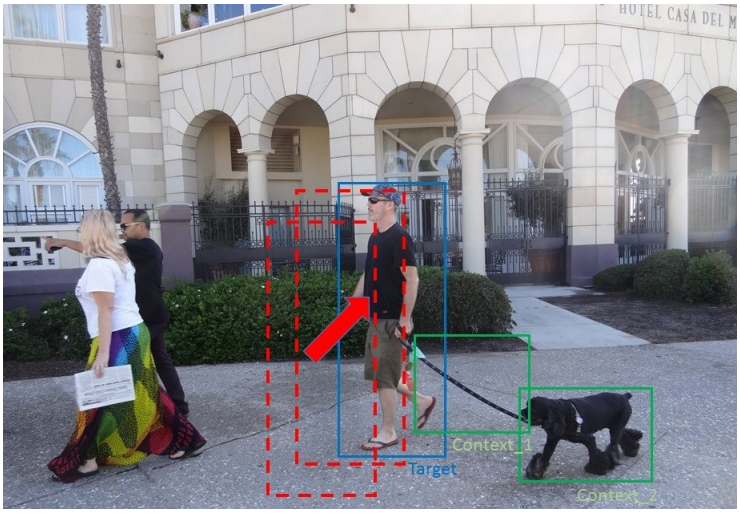


Figure 4.1: Idealization of localization process for pedestrian image using contextual data. Contextual data is shown in green; the ground-truth of the target is shown in blue, and target proposals are in red. Beginning with context-supported initial proposals, the GP-CL algorithm efficiently refines the localization process.

tion.

While the various R-CNN models perform well on general detection tasks, R-CNN-based approaches nonetheless suffer from at least four serious shortcomings and challenges: (1) the efficiency of the region proposal method, (2) the computational cost of evaluating the deep CNN, (3) localization accuracy and (4) the ability to successfully calibrate the R-CNN framework with top-down information, including context and feedback, in a principled, Bayesian manner. I address each of these four areas by proposing a Bayesian optimization scheme in conjunction with contextual visual data for efficient object localization.

This chapter provides the following contributions: (1) I demonstrate that CNN features computed from an object-proposal bounding box can be used to predict spatial offset from a target object. (2) I frame the localization process as an active search integrating top-down information in concert with a dynamic Bayesian optimization procedure requiring very few bounding-box proposals for high accuracy.

(3) By rendering an active Bayesian search, my method can provide a principled and interpretable groundwork for more complex vision tasks, which I show explicitly through the incorporation of flexible context models. I compare my approach with the bounding-box regression method used in R-CNN approaches through experiments that test efficiency and accuracy for a challenging localization task.

My primary hypothesis for this research is that object localization can be made more efficient by utilizing deep model-based object detection in combination with non-parametric Bayesian active search. To test this hypothesis, I devise an algorithm using a Bayesian optimization framework in conjunction with contextual visual data for the localization of objects in still images. My method encompasses an active search procedure that uses contextual data to generate initial bounding-box proposals for a target object. I train a convolutional neural network to approximate an offset distance from the target object. Next, I use a Gaussian Process to model this offset response signal over the search space of the target. I then employ a Bayesian active search for accurate localization of the target. In experiments, I compare my approach to a state-of-the-art bounding-box regression method for a challenging pedestrian localization task. As a validation of my initial hypothesis, I show that my method exhibits a substantial improvement over this baseline regression method.

## 4.2 Background and Related Work

My method supports a human-like approach to active object localization (e.g., [7], [15], [23]), in which a search for objects unfolds over a series of time steps. At each time step, the system uses information gained in previous time steps to decide where to search. More recent variants of R-CNN, including, notably, Faster R-CNN [32], have attempted in the main to improve the efficiency of the core R-CNN pipeline by refining either the region proposal generation stage or the proposal classification stage

of the localization algorithm. Faster RCNN trains a region-proposal network (RPN) that shares full-image convolutional features with the detection network used in Fast R-CNN [13] to simultaneously predict object bounds and objectness scores. Other related methods (e.g., [18], [36]), attempt to simplify the CNN structure to improve computation time.

I use a context-situation model, incorporating top-down, "situational" information to efficiently generate region proposals and then incorporate a Bayesian optimization scheme to further refine these proposals for accurate localization. The various R-CNN models all use category-specific "bounding-box regression" (BB-R) models to refine object proposals made by the system. The work of Zhang et al. [43] provides an extension of R-CNN that relates closely to the present work due to its use of Bayesian optimization.

Recently, contextual information has been identified to improve several vectors of analysis in computer vision, including localization [39]. Indeed, the effective use of context is critical for future A.I. systems that aim to exhibit more comprehensive capabilities, including scene and situation "understanding" [30]. Nonetheless, many current systems disregard the use of context entirely, and its apposite use in vision tasks remains an open question. Torralba and Murphy [25] incorporate global contextual features to learn context priors for object recognition. [26] frame localization as a MDP and apply unary and binary object contextual features to improve the search for a target object. Another successful use of context for localization includes [1] for which the class-specific search algorithm learns a strategy to localize objects by sequentially evaluating windows, based on statistical relation between the position and appearance of windows in the training images to their relative position with respect to the ground-truth. See also: [16], [6], [4], [28].

## 4.3 Gaussian Processes with Context-Supported Priors for Active Object Localization

Gaussian Processes used in conjunction with a Bayesian optimization framework are frequently applied in domains for which it is either difficult or costly to directly evaluate an objective function. In the case of object detection and localization, it is computationally prohibitive to extract CNN features for numerous bounding-box proposals (this is why, for instance, Faster R-CNN utilizes shared convolutional features). There consequently exists a fundamental tension at the heart of any object localization paradigm: with each bounding box for which I extract CNN features, we gain useful knowledge that can be directly leveraged in the localization process, but each such piece of information comes at a price.

A Bayesian approach is well-suited for solving the problem of function optimization under these challenging circumstances. In the case of accurate object localization, I am attempting to minimize the spatial offset from a ground-truth bounding box. To do this, I train a model to predict spatial offset of a proposal using CNN features extracted from the proposal. Once trained, the model output can be used to minimize the predicted offset. Ideally, this output is minimal when the proposal aligns with the actual ground-truth bounding box for the target object.

In my approach, I optimize a cheap approximation— the *surrogate* to the offset prediction—over the image space for efficiency. After rendering this approximation, the system determines where to sample next according to the principle *maximum expected utility*. Utility is determined using a dynamically defined acquisition function that strikes a balance between minimizing uncertainty and greedy optimization. This method is described in more detail below.

I train a model that predicts the normalized offset distance from a target ground-

truth object for a misaligned object proposal. The output of this model is the predicted distance of a proposal's center from the center of the target object, and the inverse of the output is the predicted proximity. I call the latter the "response signal." The higher the response signal, the closer the proposal is predicted to be to the target.

For each image in the training set, I generate a large number of image crops that are offset from the ground-truth pedestrian by a random amount. These randomized offset crops cover a wide range of IOU values (with respect to the ground-truth bounding box). These offset crops are also randomly scaled, so that the offset-prediction model can learn scale-invariance (with regard to bounding box size) for approximating offset distance. For each of the offset crops, I extracted CNN features using a pre-trained VGG network.

Using these features, I trained a ridge regression model mapping features to normalized offset distance from the ground-truth bounding box center. Note that in this regime, small offsets from the center of the target ground will yield (ideally) a maximum response signal. To improve the accuracy of the offset predictor, I average an ensemble of model outputs ranging over five different bounding-box scales. The performance results of the offset-prediction model are plotted in Figure 4.2.

## 4.4 Context-Situation Learning

I define a context-situation model as a distribution of location and size parameters for a target object bounding box, given various location and size parameters for a particular visual situation:

$$p(\mathbf{x}_{target}, \mathbf{s}_{target} | \{\mathbf{x}_{context}, \mathbf{s}_{context}\}_{1:C}) \tag{4.1}$$

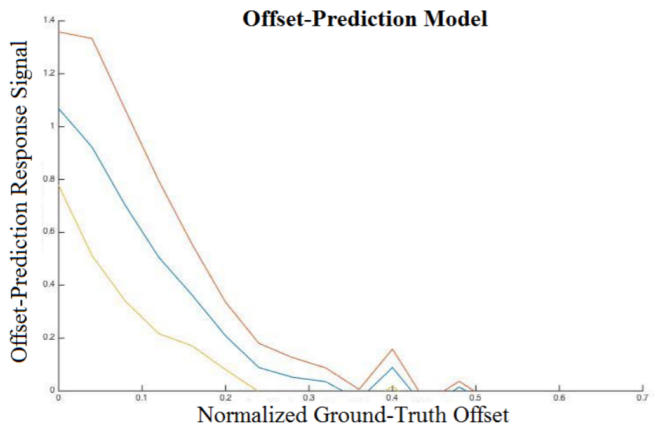


Figure 4.2: Performance of the offset-prediction model on test data (n = 1000 offset image crops). The mean (center curve) and +/-1 standard deviations (outer curves) are shown. As desired, the response signal yields a Gaussian-like peak around the center of the target object bounding-box (i.e., zero ground-truth offset). The bumps present in the range of values above 0.35 offset from the ground truth is indicative of noisy model outputs when offset crops contain no overlap with the target object.

Where $\mathbf{x} \in \mathbb{R}^2$ is the normalized bounding-box center, $\mathbf{s} \in \mathbb{R}^\mathbf{2}$ has components equal to the log bounding-box area-ratio (relative to the entire image) and log aspect-ratio, respectively; $C$ represents the number of known context objects.

In experiments, I use a set of pedestrian images for my dataset that comprise instances of a "dog-walking" visual situation.

## 4.5 Gaussian Processes

I use a Gaussian Process (GP) to compute a surrogate function f using observations $y$ of response signals from my prediction model: $y(x) = f_0(x) + \epsilon$. (Recall that the signal $y$ is high when the input proposal is predicted to be close to the target object.) The surrogate function approximates $f_0$, the objective signal value for coordinates $x$ in the image space, with $\epsilon$ connoting the irreducible error for the model.

GPs offer significant advantages over other general-purpose approaches in supervised learning settings due in part to their non-parametric structure, relative ease of

computation and the extent to which they pair well with a Bayesian modeling regime. GPs have been applied recently with success in a rich variety of statistical inference domains.

More formally, let $x_i \in R^2$ be the $i$th observation from a dataset: $D_{1:T} = \{x_{1:T}, y(x_{1:T})\}$ consisting of total pairs of object-proposal coordinates in the image space and response signals $y$, respectively. I wish to estimate the posterior distribution $p(f|D_{1:T})$ of the objective function given these data: $p(f|D_{1:T}) \propto p(D_{1:T}|f)p(f)$. This simple formula allows us to iteratively update the posterior over the signal as I acquire new data.

A GP for regression defines a distribution over functions with a joint Normality assumption. Denote $f$, the realization of the Gaussian process:

$$f \sim GP(m, k) \tag{4.2}$$

Here the GP is fully specified by the mean $m$ and covariance $k$. A common kernel function that obeys suitable continuity characteristics for the GP realization is the squared-exponential kernel, which I use here:

$$k(x, \acute{x}) = \sigma_f^2 exp\big[ -\frac{1}{2l^2}\|x - \acute{x}\|^2\big] + \sigma_\epsilon^2 \delta_{x\acute{x}} \tag{4.3}$$

where $\sigma_f^2$ is the variance of the GP realization, which I set heuristically; $\sigma_\epsilon^2$ is the variance of the $\epsilon$ parameter that I estimate empirically; and $\delta_{x\acute{x}}$ is the Kroenecker delta function. GPs are particularly sensitive to the choice of the length-scale/bandwidth parameter $l$, which I optimize with grid search for the reduced log marginal likelihood.

The posterior predictive of the surrogate function for a new datum is given by:

$$p(f_*|x_*, X, y) = N(f_*|k_*^T K_\sigma^{-1} y, k_{**} - k*^T K_\sigma^{-1} k_*) \tag{4.4}$$

where $X$ the data matrix (all prior observations $x$), $k_* = [k(x_*, x_1), \ldots, k(x_*, x_T)]$, $k_{**} = k(x_*, x_*)$ and $K_\sigma = K + \sigma_y^2 I_T$, where $K = k(x_i, x_j)$, $1 \leq i,j \leq T$.

For my algorithm, I compute posterior predictive updates in batch iterations. At each iteration, the realization of the GP is calculated over a grid of size $M$ corresponding with the image space domain of the object localization process. This grid size can be chosen to match a desired granularity/computational overhead tradeoff.

I furthermore incorporate a "short memory" mechanism in my final algorithm so that older proposal query values, which convey less information pertinent to the current localization search, are "forgotten." For improved numerical stability, I apply a Cholesky decomposition prior to matrix inversion.

### 4.6 Bayesian Optimization for Active Search

In the regime of Bayesian optimization, acquisition functions are used to guide the search for the optimum of the surrogate approximating the true objective function. Intuitively, acquisition functions are defined in such a way that *high acquisition* indicates greater likelihood of an objective function optimum. Most commonly, acquisition functions encapsulate a data query experimental design that favors either regions of large signal response, large uncertainty, or a combination of both.

At each iteration of the current algorithm, the acquisition function, defined below, is maximized to determine where to sample from the objective function (i.e., the response signal value) next. The acquisition function incorporates the mean and variance of the predictions over the image space to model the utility of sampling. The

system then evaluates the objective function at these maximal points and the Gaussian process is updated appropriately. This procedure is iterated until the stopping condition is achieved.

A standard acquisition function used in applications of Bayesian optimization is the *Expected Improvement* (EI) function. I define a dynamic variant of EI that I call Confidence-EI (CEI) that better accommodates the current problem setting:

$$a_{CEI} = \begin{cases} (\mu(x) - f(x^+) - \xi)\phi(Z) + \sigma(x)\varphi(Z) \\ Z = \frac{\mu(x) - f(x^+) - \xi}{\sigma(x)} \end{cases} \tag{4.5}$$

Above, $f(x^+)$ represents the incumbent maximum of the surrogate function, $\mu(x)$ is the mean of the surrogate at the input point x in the image space, $\sigma(x) > 0$ is the standard deviation of the surrogate at the input; $\varphi(\cdot)$ and $\phi(\cdot)$ are the pdf and cdf of the Gaussian distribution, respectively; and $\xi$ is the dynamically-assigned design parameter. The design parameter controls the exploration-exploitation tradeoff for the Bayesian optimization procedure; if, for instance, I set $\xi = 0$, then EI performs greedily.

For my algorithm, I let $\xi$ vary over the course of localization run by defining it as a function of a per-iteration total confidence score. With each iteration of localization, I set the current total confidence value equal to the median of the response signal for the current batch of bounding-box proposals. In this way, high confidence disposes the search to be greedy and conversely low confidence encourages exploration.

## 4.7 GP-CL Algorithm

**Input**: Image $I$, a set of $C$ context objects, trained model $y$ giving response signals, learned context-situation model $p(x_{target}, s_{target}|\cdot)$, $n_0$ initial bounding-box proposals

for target object generated by the context-situation model, and corresponding response signal values: $D_{n_0} = (x_i, s_i), y(x_i, s_i)_{i=1}^{n_0}$, GP hyperparameters $\theta$, size of GP realization space $M$, dynamic design parameter for Bayesian active search $\xi$, size of GP memory $GP_{mem}$ (as number of generations used), batch size $n$, number of iterations $T$, current set of bounding-box proposals and response signals $D_{proposal}^{(t)}$.

**Algorithm 4.1:** GP-CL Algorithm

Compute $n_0$ initial bounding box proposals: $\{(x_i, s_i\}_{i=1}^{n_0} \sim p(x_{target}, s_{target}|\cdot)$
$D_{proposal}^{(0)} \leftarrow D_{n_0}$
**for** $t = 1$ to $T$ **do**
    Compute $\mu(x)^{(t)}$ and $\sigma(x)^{(t)}$ for the GP realization $f_M^{(t)}$ of $D_{proposal}^{(t-1)}$ over grid of $M$ points
    **for** $i = 1$ to $n$ **do**
        $z_i = \text{argmax}_x \left(a_{CEI}(f_M^{(t)} \ \{z_j\}_{j=1}^{j=i-1}, \xi\right)$
        $sample : s_i \sim p(\cdot)_s$
        $p_i = (z_i, s_i)$
    **end for**
    $D^{(t)} \leftarrow (x_i, s_i), y(x_i, s_i)_{i=1}^{n}$
    $D_{proposal}^{(t)} \leftarrow \bigcup_{j=t-GP_{mem}}^{t} D^{(j)}$
**end for**
**return** $\max_x \ \mu(x)^{(T)}$

## 4.8 Experimental Results

I evaluate the GP-CL algorithm described above in comparison with the benchmark bounding-box regression model used in Faster R-CNN for the task of pedestrian localization.

The output of the GP-CL algorithm is a single bounding-box, as in the case of the regression model. For each method, I compare the final bounding-box with the ground-truth for the target object. In total, I tested each method for 440 experimental trials, including multiple runs with different initializations on test images.

Girshick et al. thresholded their training regime for localization with bounding-

box regression at large bounding-box overlap (IOU $\geq 0.6$). To comprehensively test my method against bounding-box regression (BB-R), I trained two distinct regression models: one with IOU thresholded for training at 0.6, as used with R-CNN, and one with IOU thresholded at 0.1.

Results for my experiments are summarized in Table 4.1 and Figure 4.4. I report the median and standard error (SE) for IOU difference (final – initial), the median relative IOU improvement (final – initial) / initial, the total percentage of the test data for which the method yielded an IOU improvement, in addition to the total percentage of test data for which the target was successfully localized (i.e., final IOU $\geq 0.5$).

| Method | IOU Difference Median (SE) | Median Relative IOU Improvement | %of Test Set with IOU Improvement | % of Test Set Localized |
|---|---|---|---|---|
| BB-R(0.6) | .1065(.004) | 32.35% | **93.86**% | 48.2% |
| BB-R(0.1) | .1034(.009) | 29.0% | **71.1**% | 44.1% |
| **GP-CL** | **.4938**(.012) | **134.7**% | 87.1% | **75.7**% |

Table 4.1: Summary statistics for the pedestrian localization task. BB-R (0.6) indicates the bounding-box regression model with training thresholded at initial IOU 0.6 and above; BB-R (0.1) denotes the bounding-box regression model with training thresholded at initial IOU 0.1 and above; GP-CL denotes Gaussian Process Context Localization.

This work has provided the following research contributions:(1) I have proposed and tested a novel technique for efficient object localization combining deep learning and non-parametric Bayesian optimization. (2) This procedure involved a novel deep training for "offset" distance. (3) In experiments, my GP-CL algorithm exhibited an improvement upon the bounding-box regression technique that underlines state of the art R-CNN localization frameworks.

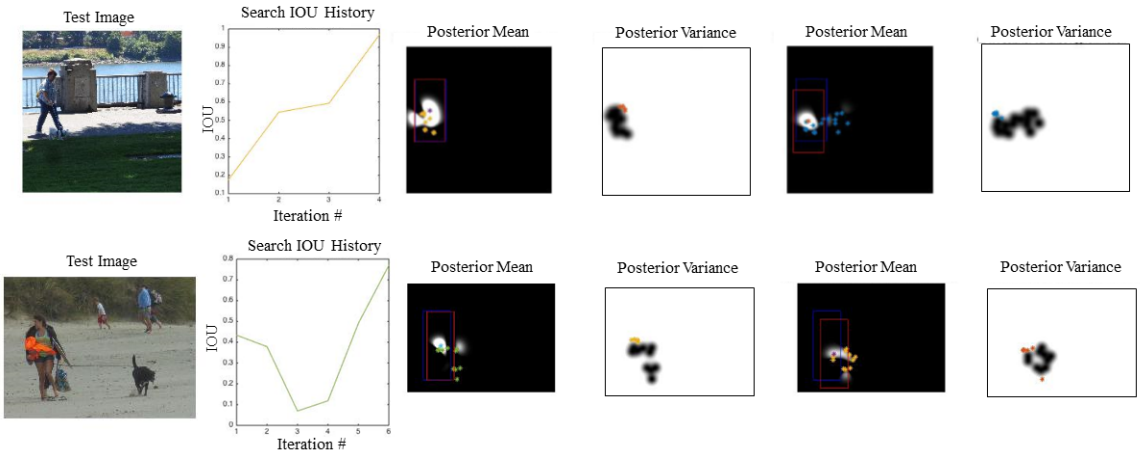


Figure 4.3: Examples of runs on two test images with the GP-CL algorithm. In each row the test image is shown on the far-left; the "search IOU history" is displayed in the second column, with the algorithm iteration number on the horizontal axis and IOU with the ground-truth target bounding box on the vertical axis. The remaining columns present the GP-CL response surface for the posterior mean and variance; the first pair of boxes reflect the second iteration of the algorithm and the last pair show the third iteration of the algorithm. In each case localization occurs rapidly thus requiring a very small number of proposals.

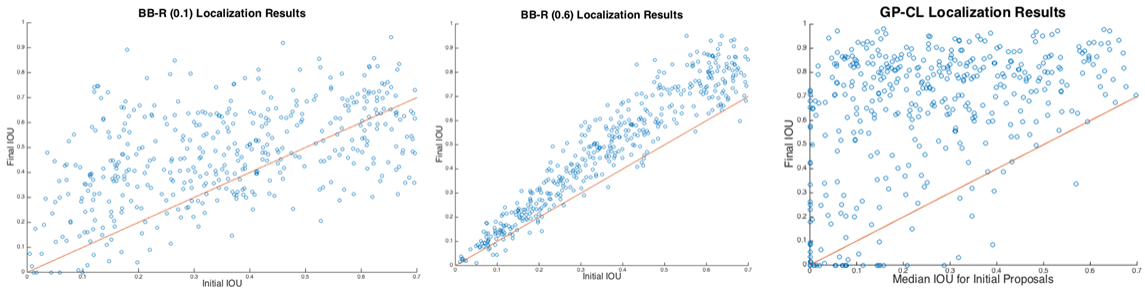


Figure 4.4: Graph of BB-R (0.6), BB-R (0.1) and GP-CL localization results for test images. The horizontal axis indicates the median IOU for the initial proposal bounding boxes, while the vertical axis designates the final IOU with the target object ground truth. The line depicted indicates "break-even" results.

# Chapter 5

# Deep Siamese Networks with Bayesian non-Parametrics for Video Object Tracking

## 5.1 Overview

In this chapter, I present a novel algorithm utilizing a deep Siamese neural network as a general object similarity function in combination with a Bayesian optimization (BO) framework to encode spatio-temporal information for efficient object tracking in video. In particular, I treat the video tracking problem as a dynamic (i.e. temporally-evolving) optimization problem. Using Gaussian Process priors, I model a dynamic objective function representing the location of a tracked object in each frame. By exploiting temporal correlations, the proposed method queries the search space in a statistically principled and efficient way, offering several benefits over current state of the art video tracking methods.

## 5.2 Background and Related Work

Early video tracking approaches have included feature-based approaches and template matching algorithms [88] that attempt to track specific features of an object or even the object as a whole. Feature-based approaches use local features, including points and edges, keypoints [89], SIFT features [90], HOG features [91] and deformable parts [92]. Conversely, template-based methods take the object as a whole

offering the potential advantage that they treat complex templates or patterns that cannot be modeled by local features alone. Through the course of a video, an object can potentially undergo a variety of different visual transformations, including rotation, occlusion, changes in scale, illumination changes, etc., that pose significant challenges for tracking. In order to obtain a robust template matching for video tracking, researchers have developed a host of methods, including mean-shift [93] and cross-correlation filtering which entails convolving a template over a search region; significant advances to cross-correlation filtering for video tracking include MOSSE [94] adaptive correlation filter and the MUSTer algorithm [95] which draws influence from cognitive psychology in the design of a flexible object representation using long and short-term memory stored by means of an integrated correlation filter.

More recently, deep learning models have been applied to video tracking to leverage the benefits of learning complex functions from large data sets. Several contemporary state of the art deep learning-based tracking models have been developed as generic object trackers in an effort to obviate the need for online training and to also improve the generalizability of the tracker. [98] applies a regression-based approach to train a generic tracker, GOTURN, offline to learn a generic relationship between appearance and motion; several deep techniques additionally incorporate motion and occlusion models, including particle filtering methods [99] and optical flow [100]. [101] demonstrated the power of deep Siamese networks based on [102], achieving a new state of the art generic object matching for video tracking.

Even with these recent successes in video object tracking, there nevertheless exists a void in state of the art video tracking workflows that fully integrate deep learning models with classical statistics and machine learning approaches. Most state of the art video trackers lack for instance a capacity to generate systematic belief states (e.g. through explicit error and uncertainty measures), or ways to seamlessly incor-

porate contextual and scene structure, or to adaptively encode temporal information (e.g. by imposing intelligent search stopping conditions and bounds) and the ability to otherwise directly and inferentially control region proposal generation or sampling methods in a precise and principled way. To this end, I wish to test whether the fusion of deep models with non-parametric approaches can provide a necessary incubation for intelligent computer vision systems capable of high-level vision tasks (e.g. efficient tracking). In the current work I present the first integrated dynamic Bayesian optimization framework in conjunction with deep learning for object tracking in video.

I adopt the Siamese network-based approach for one-shot image recognition to learn a generic, deep similarity function for object tracking. The network learns a function $f(z, x)$ that compares an exemplar crop $z$ to a candidate crop $x$ and returns a high score if the two images depict the same object and a low score otherwise. For computer vision tasks, a natural candidate for the similarity function $f$ is a deep conv-net. A Siamese network applies an identical transformation $\phi$ to both input image crops and then combines their representations using another function g that is trained to learn a general similarity function on the deep conv-net features, so that $f(z, x) = g(\phi(z), \phi(x))$.

The network is trained on positive and negative pairs, using logistic loss:

$$l(y, v) = \log(1 + \exp(-yv)) \tag{5.1}$$

where $v$ is the real-valued score of an exemplar-candidate pair and $y \in \{-1, +1\}$ is its ground-truth label. The parameters of the conv-net $\theta$ are obtained by applying Stochastic Gradient Descent (SGD) to:

$$\arg\min_{\theta} \mathbb{E}_{(z,x,y)}[L(y, f(z, x; \theta))] \tag{5.2}$$

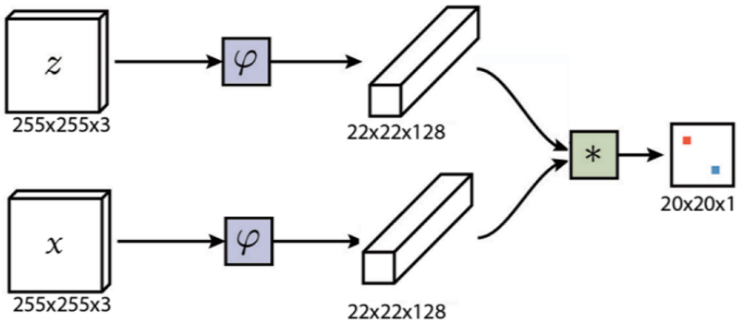


Figure 5.1: The Siamese network $\phi$ takes the exemplar image $z$ and search image $x$ as inputs. I then convolve (denoted by *) the output tensors to generate a similarity score. Similarity scores for a batch of sample search images are later rendered in a $20 \times 20 \times 1$ search grid using a Gaussian Process (see section 5.3 for details).

where the expectation is computed over the data distribution. Pairs of image crops are obtained using annotated videos from the 2015 edition of ImageNet for Large Scale Visual Recognition Challenge (ILSVRC); images are extracted from two different frames, at most a distance of T frames apart; positive image exemplars are defined as a function of their center offset distance from the ground-truth and the network stride length. Image sizes are normalized for consistency during training. I use a five-layer conv-net architecture, with pooling layers after the first and second layers, and stride lengths of 2 and 1 throughout. The final network output is a $22 \times 22 \times 128$ tensor, as shown in Figure 5.1.

### 5.3 Dynamic Bayesian Optimization

Following [113], I define object tracking in video as a dynamic optimization problem (DOP):

$$DOP = \{\max\ f(\boldsymbol{x}, t)\ s.t.\ x \in F(t) \subseteq \mathcal{S}, t \in \mathcal{T}\} \tag{5.3}$$

where $\mathcal{S} \in \mathbb{R}^D$, with $\mathcal{S}$ in the search space; $f : \mathcal{S} \times \mathcal{T} \rightarrow\ \mathbb{R}$ is the temporally-evolving objective function which yields a maximum when the input $\boldsymbol{x}$ matches the ground-

truth of the target object; $F(t)$ is the set of all feasible solutions $\boldsymbol{x} \in F(t) \subseteq \mathcal{S}$ at time $t$.

I devise a novel acquisition function, which I call *memory-score expected-improvement* (MS-EI), that demonstrated superior performance to EI and PI on my experimental data. I define MS-EI as:

$$MS\text{-}EI(\boldsymbol{x}) = (\mu(\boldsymbol{x}) - f(\boldsymbol{x}*) - \xi)\Phi(Z) + \sigma(\boldsymbol{x})\rho(Z) \tag{5.4}$$

where $Z = \frac{\mu(\boldsymbol{x}) - f(\boldsymbol{x}*) - \xi}{\sigma(\boldsymbol{x})}, \boldsymbol{x}* = \text{argmax } f(\boldsymbol{x}), \Phi$ and $\rho$ denote the pdf and xdf of the standard normal distribution respectively. I define $\xi = (\alpha \cdot \text{mean}[f(x)]_D \cdot n^q)^{-1}$; where $\alpha$ and $q$ are tunable parameters that depend on the scale of the objective function (I use $\alpha = 1, q = 1.1$); $D$ denotes the sample data set, and $n$ is the sample iteration number, with $|D| = n$; $\text{mean}[f(x)]_D$ is the sample mean of the previously observed values. Here $\xi$ serves to balance the exploration-exploitation trade-off to the specificity of a particular search. In this way, MS-EI employs a cooling schedule so that exploration is encouraged early in the search; however, the degree of exploration is conversely dynamically attenuated for exploitation as the search generates sample points with larger output values.

## 5.4 Dynamic Gaussian Processes

I model a DOP $f(\boldsymbol{x}, t)$ as a spatio-temporal GP where the objective function at time t represents a slice of f constrained at t. This dynamic GP model will therefore encapsulate statistical correlations in space and time; furthermore, the GP can enable tracking the location of an object, expressed as the temporally-evolving maximum of the objective function $f(\boldsymbol{x}, t)$.

Let $\hat{f}(\boldsymbol{x}, t) \sim \mathcal{GP}(0, K(\{\boldsymbol{x}, t\}, \{\boldsymbol{x}, t\}))$, where $(\boldsymbol{x}, t) \in \mathbb{R}^3$ ($\boldsymbol{x}$ is the bounding-box

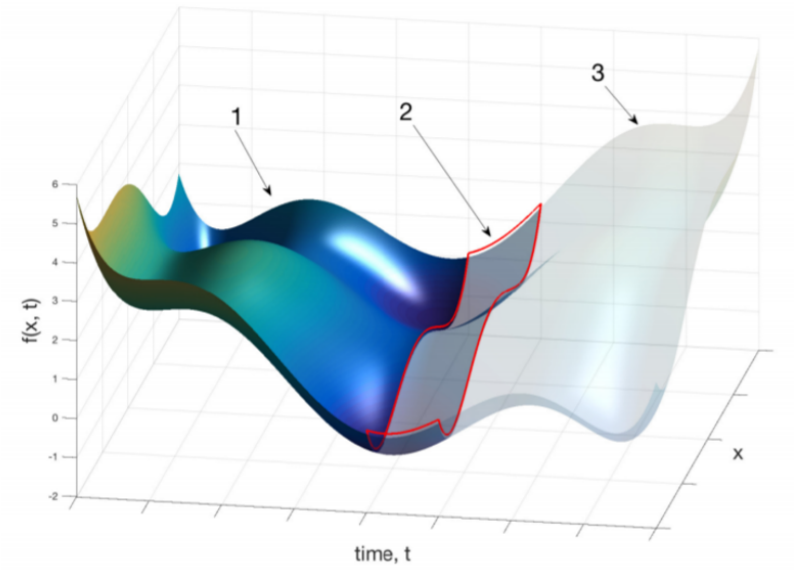


Figure 5.2: Illustration of $\hat{f}(\boldsymbol{x}, t)$ for DOP: Region (1) shows previous sample instances for time instances prior to time $t$; region (2) depicts the bounded region of the search at time $t$; region (3) represents future time slices. Image credit: [113].

spatial location), and $K$ is the covariance function of the zero-mean spatio-temporal GP. For simplicity, I assume that $K$ is both stationary and separable of the form:

$$K(\hat{f}(\boldsymbol{x}, t), \hat{f}(\boldsymbol{x}, t)) = K_{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{x}) \cdot K_T(t, t) \tag{5.5}$$

where $K_s$ and $K_T$ are the spatio and temporal covariance functions, respectively. I use Mateŕn kernel functions [119] in experiments and train the spatial and temporal covariance functions independently, following my separable assumption.

## 5.5 Siamese-Dynamic Bayesian Tracking Algorithm

I now present the details of the *Siamese-Dynamic Bayesian Tracking Algorithm* (SDBTA). The algorithm makes use of the previously-described deep Siamese conv-net. In the first step, I train the dynamic GP model. Then, for each current frame $t$ in the video containing $T$ total frames (consider $t = 0$ the initial frame containing

the ground-truth bounding-box for the target object), the algorithm renders the GPR approximation over a resized search grid of size $d \times\ d$ (I use $d = 20$ for computational efficiency), and then subsequently applies upscaling (e.g. cubic interpolation) over the original search space dimensions. In order to allow my algorithm to handle changes in the scale of the target object, each evaluation of an image crop is rendered by the Siamese network as a triplet score, where the system generates the similarity score for the current crop compared to the exemplar at three scales: $\{1.00 - p, 1.00, 1.00 + p\}$, where I heuristically set $p = 0.05$. The remaining algorithm steps are straightforward and detailed below.

**Algorithm 5.1:** Siamese-Dynamic Bayesian Tracking Algorithm

```
Train Dynamic GP model
for i do = 1,2,...T frames do
    for j do = 1,2,...{Max iterations per frame} do
        Calculate {x_i, t_i} = arg max_{x,t} MS-EI(x, t)
        Query Siamese network y_i ← f(x_i, t_i)
        Augment new point to the data
        Render GPR with set {y} over d × d grid
        Upsample grid data to dim. of search space S
        Update current location of optimum over S
    end for
end for
```

## 5.6 Experimental Results

I tested the SDBTA using a subset of the VOT14 and VOT16 datasets, the "CFNET" video tracking dataset, against three baseline video tracking models: template matching using normalized cross correlation (TM) the MOSSE tracker algorithm, and ADNET (2017, CVPR), a state of the art, deep reinforcement learning-based video tracking algorithm. During exection, I fixed the number of samples per frame at 80 (cf. region proposal systems commonly rely on thousands of image queries. I report the search summary statistics for IOU (intersection over union) for each model. Beyond

| | TM | MOSSE | ADNET | SDBTA (mine) |
|---|---|---|---|---|
| mean IOU | 0.26 | 0.10 | 0.47 | **0.56** |
| std IOU | 0.22 | 0.25 | 0.23 | **0.17** |

Table 5.1: Experimental results summary.

these strong quantitative tracking results, I additionally observed that the comparison models suffered from either significant long-term tracking deterioration or episodic instability (see Figure 5.3). The SDBTA algorithm in general did not exhibit this behavior based on my experimental trials.

This work has provided the following research contributions: (1) I propose and test a novel algorithm combining the benefits of one-shot deep learning with non-parametric Bayesian optimization. This method represents the first integrated dynamic Bayesian optimization framework in conjunction with deep learning for object tracking in video.(2) I define a novel acquisition function using a "memory" component appropriate to the task of video tracking. (3) In experiments, I show the improved performance of the SDBTA algorithm in comparison to three benchmark tracking algorithms.

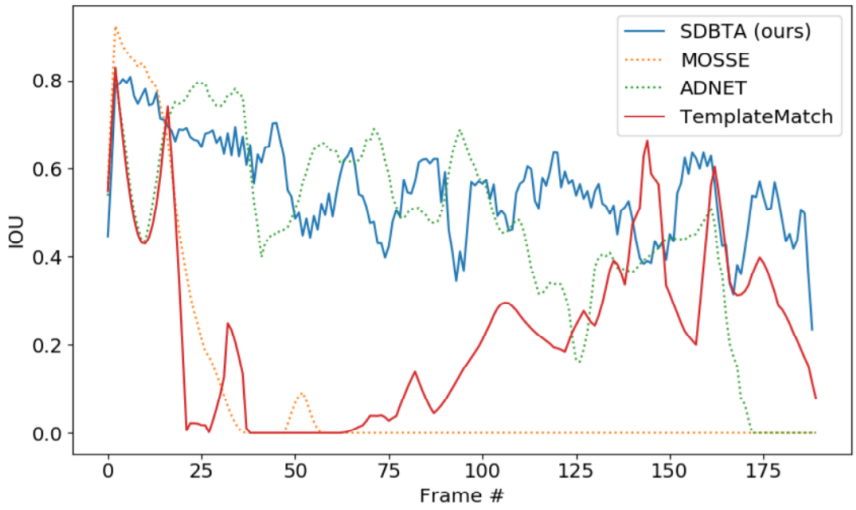


Figure 5.3: The graph shows the general stability of the SDBTA tracker for a representative test video,'tc-boat-ce1' ($T = 200$ frames); IOU is represented by the vertical axis and the frame number corresponds with the horizontal axis. By comparison, the MOSSE tracker essentially fails to track after frame 30; TM fails to track for nearly half of the duration of the video (frames 25-100); and ADNET fails to track after frame 170.

# Chapter 6

# Regularized L21-Based Semi-NonNegative Matrix Factorization with Applications to Deep Model Compression

## 6.1 Overview

In this chapter, I propose a novel data compression algorithm, *Regularized L21 Semi-NonNegative Matrix Factorization* (L21 SNF). While my algorithm serves as a general-purpose compression algorithm, I have nevertheless designed it with the specific purpose of providing an effective means to render deep model compression [170,171]. I furthermore anticipate that this work will be pertinent to applications which require effective compression of highly overdetermined datasets (e.g. genomics [173]).

Despite their recent successes, it is well known that deep models are computationally and memory intensive models. To account for the scale, diversity and the difficulty of data from which these models learn, deep networks are often deliberately built to be overly complex and to have an excess number of parameters [146]. The overdetermined nature of these models frequently makes them immensely inefficient; a recent study has shown that as much as 75% of their parameters are redundant [147]. In addition, these over-sized models have expensive inference costs, which can severely limit the feasibility of their deployment and training in constrained environments. Since their inception, many compression methods have been developed for

deep neural networks, including various regularization techniques applied to network parameters [148,149], efficient encodings [150], PCA and SVD - related dimensionality reduction techniques [151], and sparsification [152] – to name only a small number of techniques.

My algorithm aims concretely to reduce the number of filters in the convolutional tensors of a CNN via matrix decomposition. As the convolution operation represents the fundamental bottleneck across many state of the art DL models today, the effective reduction of the depth of the convolution layers has the potential to broadly increase efficiency of CNN models by reducing their computational and memory overheads. In particular, I anticipate that algorithms which make use of pre-trained, dense feature models (e.g. AlexNet, VGG) essential to high fidelity computer vision tasks (e.g. semantic segmentation, high resolution upsampling) can benefit greatly from this technique.

## 6.2 Non-Negative Matrix Factorization

Non-negative matrix factorization (NMF) is the problem of finding a matrix factorization of a given non-negative matrix $\mathbf{X}^{m\times n}$ so that $\mathbf{X} \approx \mathbf{WH}$ for non-negative factors $\mathbf{W}^{m\times r}$ and $\mathbf{H}^{r\times n}$; compression is achieved when $r < min(m, n)$, which is to say NMF produces a rank-$r$ approximation of $\mathbf{X}$. The significance of this factorization is that it gives rise, naturally, to a parts-based decomposition of $\mathbf{X}$. One can see this clearly by considering each column of $\mathbf{W}$ as a basis element in the reduced space; the columns of $\mathbf{H}$ can be interpreted as the corresponding coordinates for each basis element that render an approximation of the columns of $\mathbf{X}$. Therefore $\mathbf{W}$ can be regarded as containing a basis that is optimized for the linear approximation of the data in $\mathbf{X}$. Since the number of basis vectors ($r$) is often relatively small, this set

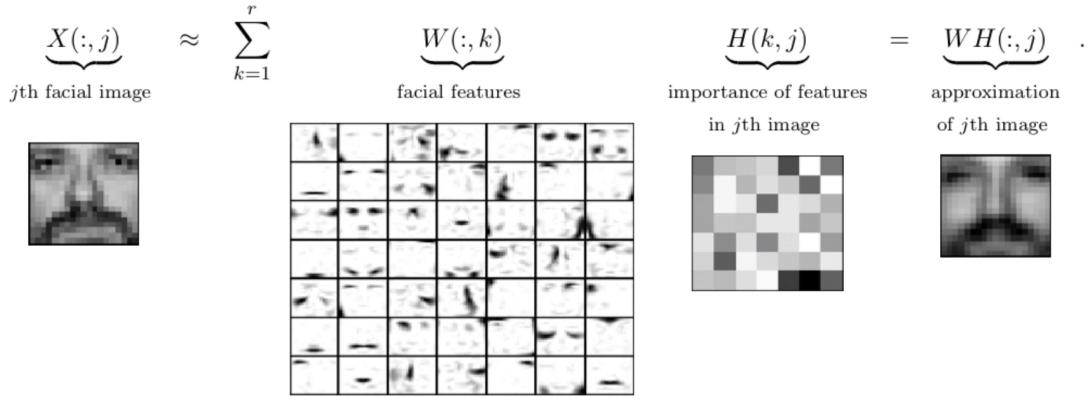


Figure 6.1: Example of parts-based NMF applied to gray-scale facial images; image credit [153].

of vectors represents a useful latent structure in the data (i.e. the matrix $\mathbf{X}$). Lastly, because each component in the factorization is restricted to be non-negative, their interaction in approximating $\mathbf{X}$ is strictly *additive*, meaning that the columns of $\mathbf{W}$ in particular yield a parts-based, compressed, decomposition of $\mathbf{X}$. See Figure 6.1 for a visualization of NMF.

To find an approximate factorization $\mathbf{X} \approx \mathbf{WH}$, I first define a cost function that quantifies the quality of the approximation. There are several natural choices. The *Frobenius norm*, an extension of the Euclidean norm to matrices and tensors provides one option:

$$\|\mathbf{X} - \mathbf{WH}\|_F^2 = \sum_{ij} (\mathbf{X}_{ij} - \mathbf{WH}_{ij})^2 \tag{6.1}$$

This expression is bounded below by zero, and vanishes when $\mathbf{X} = \mathbf{WH}$. A common alternative measure for NMF optimization is the so-called "divergence", defined:

$$D(\mathbf{X}||\mathbf{WH}) = \sum_{ij} (\mathbf{X}_{ij} \log \frac{\mathbf{X}_{ij}}{\mathbf{WH}_{ij}} - \mathbf{X}_{ij} + \mathbf{WH}_{ij}) \tag{6.2}$$

where divergence is similarly bounded below by zero, and vanishes when $\mathbf{X} = \mathbf{WH}$; divergence reduces to KL-Divergence [154] when $\sum_{ij} \mathbf{X}_{ij} = \sum_{ij} \mathbf{WH}_{ij} = 1$. In the classic paper [155], the authors solve NMF for both cost functions (6.1) and (6.2) using a multiplicative update rule; others have used non-negative least square [156], neural approaches [157] and projective methods [158].

When the data matrix $\mathbf{X}$ is not strictly non-negative (e.g. consider $\mathbf{X}$ as a convolutional tensor) NMF will fail, naturally. Nevertheless, in many common use cases, a parts-based decomposition is a desideratum for data compression with non-negative data. [163] Introduce a useful compromise toward this end, which they term "Semi-Nonnegative Matrix Factorization" in which one – and only one – of the factor matrices (i.e. $\mathbf{W}, \mathbf{H}$) is constrained to be non-negative.

In the context of model compression, I consider $\mathbf{X}$ to be a convolutional tensor, where the filters (2-d arrays) in the tensor are flattened so that the convolutional tensor as a whole admits of a matrix representation. In this manner, non-negative matrix factorization of the flattened convolutional tensor yields a parts-based decomposition that effectively reduces one or more of the tensor dimensions (e.g. the filter depth). This reduction facilitates a useful compression of a deep CNN. By analogy, this matrix factorization methodology can be extended to the domain of higher order tensors [159].

In place of the aforementioned loss functions, I propose to instead employ a generally more robust measure that combines the strengths of L2 and L1 loss, termed L2-1 loss [164]. Define the L2-1 norm as follows:

$$\|\mathbf{X}\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{m} \mathbf{X}_{ji}^2} \tag{6.3}$$

L2-1 loss is accordingly given by:

$$\|\mathbf{X} - \mathbf{WH}\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{m} (\mathbf{X}_{ji} - (\mathbf{WH})_{ji})^2} \tag{6.4}$$

*Note in particular that I define L2-1 as a sum of L2 vector magnitudes with respect to each column of $\mathbf{X}$. When applied to a set of convolutional filters (consider each column of $\mathbf{X}$ as a "flattened" filter), for example, L2-1 loss can viewed as a measure that weighs the distance per filter component using L2 cost, while summing over filters with L1 cost. One can show that the L2 norm is rotationally invariant [160] (a desirable property for approximating convolutional filters); moreover, the L1 norm is known to be robust to outliers.

## 6.3 Robust L21-Based Semi-Nonnegative Matrix Factorization

Recall the following useful gradient and trace-related formulas which I employ below:

$$tr(\mathbf{A} + \mathbf{B}) = tr(\mathbf{A}) + tr(\mathbf{B}) \tag{6.5}$$

$$tr(\mathbf{ABC}) = tr(\mathbf{CAB}) = tr(\mathbf{BCA}) \tag{6.6}$$

$$\nabla_{\mathbf{X}} tr(\mathbf{AX}) = \mathbf{A}^T \tag{6.7}$$

$$\nabla_{\mathbf{X}} tr(\mathbf{X}^T\mathbf{A}) = \mathbf{A} \tag{6.8}$$

$$\nabla_{\mathbf{X}} tr(\mathbf{X}^T\mathbf{AX}) = (\mathbf{A} + \mathbf{A}^T)\mathbf{X} \tag{6.9}$$

$$\nabla_{\mathbf{X}} tr(\mathbf{XAX}^T) = \mathbf{X}(\mathbf{A}^T + \mathbf{A}) \tag{6.10}$$

$$\|\mathbf{X}\|_2^2 = tr(\mathbf{X}^T\mathbf{X}) \tag{6.11}$$

Let $\mathbf{X} \in \mathbb{R}^{m \times n}, \mathbf{W} \in \mathbb{R}^{m \times k}, \mathbf{H} \in \mathbb{R}_{+}^{k \times n}$; moreover, let $\mathbf{x}^{(i)} \in \mathbb{R}^{m \times 1}$ denote the ith column of $\mathbf{X}$, and $\mathbf{h}^{(i)} \in \mathbb{R}_{+}^{k \times 1}$ denote the ith column of $\mathbf{H}$. The optimization problem underlying my regularized L21 semi non-negative matrix factorization algorithm is defined:

$$\arg\min_{\mathbf{W},\mathbf{H}} \|\mathbf{X} - \mathbf{WH}\|_{2,1} + \overline{\alpha}\|\mathbf{W}\|_2^2 \text{ subject to } \mathbf{H} \geq 0 \tag{6.12}$$

where $\overline{\alpha} = \frac{\alpha}{2}$ and $\alpha \geq 0$ is a given parameter, and I adopt $\frac{\alpha}{2}$ instead of $\alpha$ here in order to simplify the new algorithm and its derivation.

I define the associated loss function:

$$\begin{aligned} \mathscr{L}(\mathbf{X}, \mathbf{WH}) = tr[(\mathbf{X} - \mathbf{WH})\mathbf{D}(\mathbf{X} - \mathbf{WH})^T] + \alpha tr[\mathbf{W}^T\mathbf{W}] \\ \text{where } \mathbf{D} \in \mathbb{R}^{n \times n}, \mathbf{D}_{ii} = 1/\|\mathbf{x}^{(i)} - \mathbf{Wh}^{(i)}\|_2 \end{aligned} \tag{6.13}$$

Following the methodology introduced in [167], I subsequently derive iterative update formulas based on the loss function given in (6.13) and show that these updates incur a monotonic loss in (6.12).

$$\mathscr{L}(\mathbf{X}, \mathbf{WH}) = tr[\mathbf{XDX}^T - 2\mathbf{XDH}^T\mathbf{W}^T + \mathbf{WHDH}^T\mathbf{W}^T] + \alpha tr[\mathbf{W}^T\mathbf{W}] \tag{6.14}$$

$$= tr[\mathbf{XDX}^T] - 2tr[\mathbf{W}^T\mathbf{XDH}^T] + tr[\mathbf{WHDH}^T\mathbf{W}^T] + \alpha tr[\mathbf{W}^T\mathbf{W}] \tag{6.15}$$

Observing that:

$$\nabla_{\mathbf{W}} tr[(\mathbf{XDX}^T] = 0 \tag{6.16}$$

$$\nabla_{\mathbf{W}} tr[\mathbf{W}^T\mathbf{XDH}^T] = \mathbf{XDH}^T \tag{6.17}$$

$$\nabla_{\mathbf{W}} tr[\mathbf{WHDH}^T\mathbf{W}^T] = 2\mathbf{WHDH}^T \tag{6.18}$$

$$\nabla_{\mathbf{W}} \alpha tr[\mathbf{W}^T\mathbf{W}] = 2\alpha\mathbf{W} \tag{6.19}$$

This yields:

$$\nabla_{\mathbf{W}}\mathscr{L} = 2\mathbf{WHDH}^T - 2\mathbf{XDH}^T + 2\alpha\mathbf{W} \tag{6.20}$$

I now solve $\nabla_{\mathbf{W}}\mathscr{L} = 0$, which gives the solution:

$$\mathbf{W} = [\mathbf{XDH}^T][\alpha\mathbf{I} + \mathbf{HDH}^T]^{-1} \tag{6.21}$$

Next I prove that optimality of (6.21) by demonstrating that (6.13) is convex; I first consider $\frac{\partial\mathscr{L}}{\partial W_{ij}}$:

$$\frac{\partial\mathscr{L}}{\partial W_{ij}} = 2(\mathbf{WHDH}^T)_{ij} - 2(\mathbf{XDH}^T)_{ij} + 2\alpha(\mathbf{W})_{ij} \tag{6.22}$$

By expanding the first term on the RHS of (6.22), I have:

$$\frac{\partial\mathscr{L}}{\partial W_{ij}} = 2\sum_{l=1}^{k}\mathbf{W}_{il}(\mathbf{HDH}^T)_{lj} - 2(\mathbf{XDH}^T)_{ij} + 2\alpha(\mathbf{W})_{ij} \tag{6.23}$$

The Hessian of $\mathscr{L}$ is consequently:

$$\frac{\partial\mathscr{L}}{\partial W_{ij}\partial W_{pq}} = 2(\mathbf{HDH}^T + \alpha\mathbf{I})_{qj}\delta_{ip} \qquad 1 \leq i,p \leq m \quad 1 \leq j,q \leq k \tag{6.24}$$

Therefore, the Hessian of $\mathscr{L}$ is a block diagonal matrix with each block being $2HDH^T + 2\alpha I$. Since $2HDH^T + 2\alpha I$ is a positive definite matrix of size $k \times k$, then the Hessian of $\mathscr{L}$ is also a positive definite matrix of size $mk \times mk$. This indicates that $\mathscr{L}$ is convex. Therefore the formula given for $\mathbf{W}$ in (6.21) is optimal, as was to be shown.

The previous derivation of (6.21) and associated demonstration of optimality furnish a proof for the following Lemma. I now consider (6.21) as an iterative update rule at step $t$, where I regard $\mathbf{H}(t)$ as fixed at the time of the $t$-th update for $\mathbf{W}$, denoted by

$\mathbf{W}(t)$. Define $\mathbf{D}(t)_{ii} = 1/\|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t)^{(i)}\|_2$, which is also regarded as fixed at the time of $t$-th update for W. The iterative update for matrix $\mathbf{W}$ is given by:

$$\mathbf{W}(t+1) = [\mathbf{X}\mathbf{D}(t)\mathbf{H}(t)^T][\alpha\mathbf{I} + \mathbf{H}(t)\mathbf{D}(t)\mathbf{H}(t)^T]^{-1} \tag{6.25}$$

**Lemma 1.** Let $\mathbf{W}(t)$ and $\mathbf{W}(t+1)$ represent consecutive updates for $\mathbf{W}$ as prescribed by (6.25). Under this updating rule, the following inequality holds:

$$\begin{aligned} tr[(\mathbf{X} - \mathbf{W}(t+1)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X} - \mathbf{W}(t+1)\mathbf{H}(t))^T] + \alpha tr[\mathbf{W}^T(t+1)\mathbf{W}(t+1)] \\ \leq tr[(\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t))^T)] + \alpha tr[\mathbf{W}^T(t)\mathbf{W}(t)] \end{aligned} \tag{6.26}$$

**Proof.** The proof of Lemma 1 follows directly from the optimality of the update formula given in (6.21). □

**Lemma 2.** Following the work of [167], I show that under the update rule of (6.25), the following inequality holds where $\overline{\alpha} = \frac{\alpha}{2}$:

$$\begin{aligned} &\|\mathbf{X} - \mathbf{W}(t+1)\mathbf{H}(t)\|_{2,1} + \overline{\alpha} tr[\mathbf{W}(t+1)\mathbf{W}^T(t+1)] \\ &\qquad -(\|\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t)\|_{2,1} + \overline{\alpha} tr[\mathbf{W}(t)\mathbf{W}^T(t)]) \\ &\leq \frac{1}{2}\Big[tr[(\mathbf{X} - \mathbf{W}(t+1)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X} - \mathbf{W}(t+1)(\mathbf{H}(t))^T] \\ &\qquad -tr[(\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t))^T]\Big] \\ &\qquad +\overline{\alpha} tr[\mathbf{W}(t+1)\mathbf{W}^T(t+1)] - \overline{\alpha} tr[\mathbf{W}(t)\mathbf{W}^T(t)] \end{aligned} \tag{6.27}$$

**Proof.** Notice that:

$$\begin{aligned} tr[(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))^T)] + \alpha tr[\mathbf{W}(t)\mathbf{W}^T(t)] \\ = \sum_{j=1}^{p}\sum_{i=1}^{n}(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))_{ij}^2 D_{ii} + \alpha tr[\mathbf{W}(t)\mathbf{W}^T(t)] \\ = \sum_{i=1}^{n}\|\mathbf{x}^{(i)}-\mathbf{W}(t)\mathbf{h}(t)^{(i)}\|^2 D_{ii}(t) + \alpha tr[\mathbf{W}(t)\mathbf{W}^T(t)] \end{aligned} \tag{6.28}$$

Analogously,

$$\begin{aligned} tr[(\mathbf{X}-\mathbf{W}(t+1)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X}-\mathbf{W}(t+1)\mathbf{H}(t))^T)] + \alpha tr[\mathbf{W}(t+1)\mathbf{W}^T(t+1)] \\ = \sum_{i=1}^{n}\|\mathbf{x}^{(i)}-\mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\|^2 D_{ii}(t) + \alpha tr[\mathbf{W}(t+1)\mathbf{W}^T(t+1)] \end{aligned} \tag{6.29}$$

Consequently, the right-hand-side (RHS) of the inequality (6.27) can be expressed:

$$\begin{aligned} RHS = \frac{1}{2}\sum_{i=1}^{n}\Big(\|\mathbf{x}^{(i)}-\mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\|^2 D_{ii}(t) - \|\mathbf{x}^{(i)}-\mathbf{W}(t)\mathbf{h}(t)^{(i)}\|^2 D_{ii}(t)\Big) \\ +\overline{\alpha}tr[\mathbf{W}(t+1)\mathbf{W}^T(t+1)] - \overline{\alpha}tr[\mathbf{W}(t)\mathbf{W}^T(t)] \\ = \frac{1}{2}\sum_{i=1}^{n}\Big(\|\mathbf{x}^{(i)}-\mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\|^2 D_{ii}(t) - \frac{1}{D_{ii}(t)}\Big) \\ +\overline{\alpha}tr[\mathbf{W}(t+1)\mathbf{W}^T(t+1)] - \overline{\alpha}tr[\mathbf{W}(t)\mathbf{W}^T(t)] \end{aligned} \tag{6.30}$$

Similarly, the left-hand-side (LHS) of the inequality (6.27) can be expressed:

$$\begin{aligned} LHS = \sum_{i=1}^{n} \Big( \|\mathbf{x}^{(i)} - \mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\| - \|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t)^{(i)}\| \Big) + \\ \overline{\alpha} tr[\mathbf{W}(t+1)\mathbf{W}^T(t+1)] - \overline{\alpha} tr[\mathbf{W}(t)\mathbf{W}^T(t)] \\ = \sum_{i=1}^{n} \Big( \|\mathbf{x}^{(i)} - \mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\| - \frac{1}{D_{ii}} \Big) \\ + \overline{\alpha} tr[\mathbf{W}(t+1)\mathbf{W}^T(t+1)] - \overline{\alpha} tr[\mathbf{W}(t)\mathbf{W}^T(t)] \end{aligned} \tag{6.31}$$

Ergo,

$$\begin{aligned} LHS - RHS &= \sum_{i=1}^{n} \Big( \|\mathbf{x}^{(i)} - \mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\| - \frac{1}{2}\|\mathbf{x}^{(i)} - \mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\|^2 D_{ii}(t) - \frac{1}{2}\frac{1}{D_{ii}(t)} \Big) \\ &= \sum_{i=1}^{n} \frac{D_{ii}(t)}{2} \Big( 2\frac{\|\mathbf{x}^{(i)} - \mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\|}{D_{ii}(t)} - \|\mathbf{x}^{(i)} - \mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\|^2 - \frac{1}{D_{ii}(t)^2} \Big) \\ &= \sum_{i=1}^{n} \frac{-D_{ii}(t)}{2} \Big( \|\mathbf{x}^{(i)} - \mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\|^2 - 2\|\mathbf{x}^{(i)} - \mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\| \frac{1}{D_{ii}(t)} + \frac{1}{D_{ii}(t)^2} \Big) \\ &= \sum_{i=1}^{n} \frac{-D_{ii}(t)}{2} \Big( \|\mathbf{x}^{(i)} - \mathbf{W}(t+1)\mathbf{h}(t)^{(i)}\| - \frac{1}{D_{ii}(t)} \Big)^2 \\ &\leq 0 \end{aligned} \tag{6.32}$$

As was to be shown. □

**Theorem 1.** Updating **W** using formula (6.21) while fixing **H** yields a monotonic decrease in the objective function defined by (6.12).

**Proof.** By Lemma 1, the right hand side expression in Lemma 2:

$$\frac{1}{2}tr[(\mathbf{X}-\mathbf{W}(t+1)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X}-\mathbf{W}(t+1)(\mathbf{H}(t))^T]+\overline{\alpha}tr[\mathbf{W}(t+1)\mathbf{W}^T(t+1)]$$
$$-\frac{1}{2}tr[(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))^T]-\overline{\alpha}tr[\mathbf{W}(t)\mathbf{W}^T(t)] \tag{6.33}$$

is less than or equal to zero. So does the left hand side expression in Lemma 2:

$$\begin{aligned}\|\mathbf{X}-\mathbf{W}(t+1)\mathbf{H}(t)\|_{2,1}+\overline{\alpha}tr[\mathbf{W}(t+1)\mathbf{W}^T(t+1)]\\ -\|\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t)\|_{2,1}-\overline{\alpha}tr[\mathbf{W}(t)\mathbf{W}^T(t)]\leq 0\end{aligned} \tag{6.34}$$

Thus proving Theorem 1. □

Next I derive an iterative update formula for $\mathbf{H}$, with $\mathbf{H}\geq 0$; subsequently I prove convergence of this update rule by showing that the residual $\mathscr{L}(\mathbf{X},\mathbf{WH})$ (eq. (6.13)) is monotonically decreasing for fixed $\mathbf{W}$. Since the second term of the residual, $\alpha tr[\mathbf{W}^T\mathbf{W}]$, is fixed during the $\mathbf{H}$ update, I ignore it here.

**Definition.**

$$F(\mathbf{H})=tr[(\mathbf{X}-\mathbf{WH})\mathbf{D}(\mathbf{X}-\mathbf{WH})^T] \tag{6.35}$$

To prove this convergence, I utilize an auxiliary function, denoted $\mathscr{A}(\mathbf{H},\mathbf{H}')$ as in [155], [163].

**Definition.** $\mathscr{A}$ is an auxiliary function for $F(\mathbf{H})$ if:

$$\mathscr{A}(\mathbf{H},\mathbf{H}')\geq F(\mathbf{H}),\quad \mathscr{A}(\mathbf{H},\mathbf{H})=F(\mathbf{H}) \tag{6.36}$$

**Lemma 3.** If $\mathscr{A}$ is an auxiliary function of $F(\mathbf{H})$, then $F(\mathbf{H})$ is non-increasing under the update:

$$\mathbf{H^{t+1}} = \arg\min_{\mathbf{H}} \mathscr{A}(\mathbf{H}, \mathbf{H^t}) \tag{6.37}$$

**Proof.**

$$F(\mathbf{H^{t+1}}) \leq \mathscr{A}(\mathbf{H^{t+1}}, \mathbf{H^t}) \leq \mathscr{A}(\mathbf{H^t}, \mathbf{H^t}) = F(\mathbf{H^t}). \tag{6.38}$$

I now consider an explicit solution for $\mathbf{H}$ in the form of an iterative update, for which I subsequently prove convergence. Since $\mathbf{H}$ is non-negative, it is helpful to decompose both the $k \times k$ matrix $\mathbf{W}^T\mathbf{W} = \Omega$ and the $k \times n$ matrix $\mathbf{W}^T\mathbf{X} = \Phi$ into their positive and negative entries:

$$\Omega_{ij}^{+} = \frac{1}{2}(|\Omega_{ij}| + \Omega_{ij}), \quad \Omega_{ij}^{-} = \frac{1}{2}(|\Omega_{ij}| - \Omega_{ij}). \tag{6.39}$$

**Lemma 4.** Under the iterative update:

$$\mathbf{H}_{ij}(t+1) = \mathbf{H}_{ij}(t)\sqrt{\frac{(\Phi^{+}\mathbf{D}(t))_{ij} + (\Omega^{-}\mathbf{H}(t)\mathbf{D}(t))_{ij}}{(\Phi^{-}\mathbf{D}(t))_{ij} + (\Omega^{+}\mathbf{H}(t)\mathbf{D}(t))_{ij}}} \tag{6.40}$$

where $\mathbf{W}^T(t)\mathbf{W}(t) = \Omega$, $\Omega = \Omega^{+} - \Omega^{-}$, $\mathbf{W}^T(t)\mathbf{X} = \Phi$, $\Phi = \Phi^{+} - \Phi^{-}$, and $\mathbf{D}(t)_{ii} = 1/\|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t)^{(i)}\|_2$, the following relation holds for some auxiliary function $\mathscr{A}(\mathbf{H}, \mathbf{H}')$:

$$\mathbf{H}(t+1) = \arg\min_{\mathbf{H}} \mathscr{A}(\mathbf{H}, \mathbf{H}(t)) \tag{6.41}$$

**Proof.** Using the notation introduced above, $F(\mathbf{H})$ can be written in the following form:

$$\begin{aligned} F(\mathbf{H}) = tr[\mathbf{X}\mathbf{D}\mathbf{X}^T] - 2tr[\mathbf{H}^T\mathbf{\Phi}^{+}\mathbf{D}] + 2tr[\mathbf{H}^T\mathbf{\Phi}^{-}\mathbf{D}] \\ + tr[\Omega^{+}\mathbf{H}\mathbf{D}\mathbf{H}^T] - tr[\Omega^{-}\mathbf{H}\mathbf{D}\mathbf{H}^T] \end{aligned} \tag{6.42}$$

In the subsequent steps I provide an auxiliary function $\mathscr{A}(\mathbf{H}, \mathbf{H}')$ for the residual loss $F(\mathbf{H})$. Following [164], in order to construct an auxiliary function that furnishes an upper-bound for $F(\mathbf{H})$, I define $\mathscr{A}(\mathbf{H}, \mathbf{H})$ as a sum comprised of terms that represent upper-bounds for each of the positive terms appearing in (6.42) and lower-bounds for each of the negative terms, respectively.

Using the fact that $a \leq \frac{a^2+b^2}{2b} \forall\ a, b > 0$, I derive an upper bound for the third term on the RHS of (6.42):

$$tr[\mathbf{H}^T \mathbf{\Phi}^- \mathbf{D}] = \sum_{ij} \mathbf{H}_{ij} (\mathbf{\Phi}^- \mathbf{D})_{ij} \leq \sum_{ij} (\mathbf{\Phi}^- \mathbf{D})_{ij} \frac{(\mathbf{H}_{ij})^2 + (\mathbf{H}'_{ij})^2}{2\mathbf{H}'_{ij}} \tag{6.43}$$

[163] derive the following useful inequality, which I use to bound the fourth term on the RHS of (6.42).

**Proposition 1.** For any matrices $\mathbf{A} \in \mathbb{R}_+^{n \times n}$, $\mathbf{B} \in \mathbb{R}_+^{k \times k}$, $\mathbf{S} \in \mathbb{R}_+^{n \times k}$, $\mathbf{S}' \in \mathbb{R}_+^{n \times k}$, with $\mathbf{A}$ and $\mathbf{B}$ symmetric:

$$tr[\mathbf{S}^T \mathbf{A} \mathbf{S} \mathbf{B}] \leq \sum_{i=1}^{n} \sum_{p=1}^{k} \frac{(\mathbf{A}\mathbf{S}'\mathbf{B})_{ip} \mathbf{S}_{ip}^2}{\mathbf{S}'_{ip}} \tag{6.44}$$

Considering the fourth term of the RHS of (28), I have:

$$tr[\Omega^+ \mathbf{H} \mathbf{D} \mathbf{H}^T] = tr[\mathbf{H}^T \Omega^+ \mathbf{H} \mathbf{D}] \leq \sum_{ij} \frac{(\Omega^+ \mathbf{H}' \mathbf{D})_{ij} \mathbf{H}_{ij}^2}{\mathbf{H}'_{ij}} \tag{6.45}$$

Next I derive a lower-bound for the second term of (6.42), using the fact that $a \geq$

$1 + log\ a,\ \forall\ a > 0$:

$$\begin{aligned} tr[\mathbf{H}^T\mathbf{\Phi}^+\mathbf{D}] &= \sum_{ij} \mathbf{H}_{ij}(\mathbf{\Phi}^+\mathbf{D})_{ij} \\ &\geq \sum_{ij} (\mathbf{\Phi}^+\mathbf{D})_{ij}\mathbf{H}'_{ij}(1 + log\frac{\mathbf{H}_{ij}}{\mathbf{H}'_{ij}}) \end{aligned} \tag{6.46}$$

Finally, I consider the last term on the RHS of equation (6.42):

**Proposition 2.**

$$tr[\Omega^-\mathbf{H}\mathbf{D}\mathbf{H}^T] \geq \sum_{ijk} \Omega^-_{ik}\mathbf{H}'_{kj}\mathbf{D}_{jj}\mathbf{H}'_{ij}\Big(1 + log\frac{\mathbf{H}_{kj}\mathbf{H}_{ij}}{\mathbf{H'}_{kj}\mathbf{H}'_{ij}}\Big) \tag{6.47}$$

**Proof.**

$$\begin{aligned} tr[\Omega^-\mathbf{H}\mathbf{D}\mathbf{H}^T] = tr[\mathbf{H}^T\Omega^-\mathbf{H}\mathbf{D}] &= \sum_{ij} (\Omega^-\mathbf{H}\mathbf{D})_{ij}\mathbf{H}_{ij} \\ = \sum_{ijk} \Omega^-_{ik}(\mathbf{H}\mathbf{D})_{kj}\mathbf{H}_{ij} &= \sum_{ijk} \Omega^-_{ik}\mathbf{H}_{kj}\mathbf{D}_{jj}\mathbf{H}_{ij} \end{aligned} \tag{6.48}$$

Once again I employ the inequality $a \geq 1 + log\ a$, whereupon:

$$tr[\Omega^-\mathbf{H}\mathbf{D}\mathbf{H}^T] \geq \sum_{ijk} \Omega^-_{ik}\mathbf{H}'_{kj}\mathbf{D}_{jj}\mathbf{H}'_{ij}\Big(1 + log\frac{\mathbf{H}_{kj}\mathbf{H}_{ij}}{\mathbf{H'}_{kj}\mathbf{H}'_{ij}}\Big) \tag{6.49}$$

As was to be shown. □

Putting the previous steps together, I define the auxiliary function $\mathscr{A}(\mathbf{H}, \mathbf{H}')$:

$$\begin{aligned}\mathscr{A}(\mathbf{H}, \mathbf{H}') = tr[\mathbf{X}\mathbf{D}\mathbf{X}^T] + \sum_{ij} \frac{(\Omega^+\mathbf{H}'\mathbf{D})_{ij}\mathbf{H}_{ij}^2}{\mathbf{H}'_{ij}} - 2\sum_{ij}(\mathbf{\Phi}^+\mathbf{D})_{ij}\mathbf{H}'_{ij}(1 + log\frac{\mathbf{H}_{ij}}{\mathbf{H}'_{ij}}) \\ - \sum_{ijk}\Omega^-_{ik}\mathbf{H}'_{kj}\mathbf{D}_{jj}\mathbf{H}'_{ij}\Big(1 + log\frac{\mathbf{H}_{kj}\mathbf{H}_{ij}}{\mathbf{H}'_{kj}\mathbf{H}'_{ij}}\Big) + 2\sum_{ij}(\mathbf{\Phi}^-\mathbf{D})_{ij}\frac{(\mathbf{H}_{ij})^2 + (\mathbf{H}'_{ij})^2}{2\mathbf{H}'_{ij}}\end{aligned} \tag{6.50}$$

Observe that $\mathscr{A}(\mathbf{H}, \mathbf{H}') \geq F(\mathbf{H})$ and $\mathscr{A}(\mathbf{H}, \mathbf{H}) = F(\mathbf{H})$, as required for an auxiliary function, where $F(\mathbf{H})$ denotes the residual loss as defined in equation (6.42). By the aforementioned Lemma, it follows that $F(\mathbf{H})$ is non-increasing under the update: $\mathbf{H}(t+1) = \arg\min_{\mathbf{H}} \mathscr{A}(\mathbf{H}, \mathbf{H}(t))$.

I now demonstrate that the minimum of $\mathscr{A}(\mathbf{H}, \mathbf{H}')$ coincides with the update rule given in (6.40), by: (1) showing that the update in (6.40) corresponds with a critical point for $\mathscr{A}(\mathbf{H}, \mathbf{H}')$; and (2) proving the convexity of $\mathscr{A}(\mathbf{H}, \mathbf{H})$. Since

$$\begin{aligned}\frac{\partial\mathscr{A}(\mathbf{H}, \mathbf{H}')}{\partial\mathbf{H}_{ij}} = 2(\mathbf{\Phi}^-\mathbf{D})_{ij}\Big(\frac{\mathbf{H}_{ij}}{\mathbf{H}'_{ij}}\Big) + 2\frac{(\Omega^+\mathbf{H}'\mathbf{D})_{ij}\mathbf{H}_{ij}}{\mathbf{H}'_{ij}} \\ -2(\mathbf{\Phi}^+\mathbf{D})_{ij}\Big(\frac{\mathbf{H}'_{ij}}{\mathbf{H}_{ij}}\Big) - 2\frac{(\Omega^-\mathbf{H}'\mathbf{D})_{ij}\mathbf{H}'_{ij}}{\mathbf{H}_{ij}} = 0\end{aligned} \tag{6.51}$$

Solving for $\mathbf{H}_{ij}$ gives:

$$\mathbf{H}_{ij} = \mathbf{H}'_{ij}\sqrt{\frac{(\Phi^+\mathbf{D})_{ij} + (\Omega^-\mathbf{H}'\mathbf{D})_{ij}}{(\Phi^-\mathbf{D})_{ij} + (\Omega^+\mathbf{H}'\mathbf{D})_{ij}}} \tag{6.52}$$

Thus, as asserted, the update rule given in (6.40) corresponds with a critical point for $\mathscr{A}(\mathbf{H}, \mathbf{H}')$.

Now I consider computation of the Hessian of $\mathscr{A}(\mathbf{H}, \mathbf{H}')$:

$$\frac{\partial \mathscr{A}(\mathbf{H}, \mathbf{H}')}{\partial \mathbf{H}_{ij} \partial \mathbf{H}_{kl}} = \begin{cases} \text{if } (i,j) == (k,l): \\ 2\frac{(\Phi^- \mathbf{D})_{ij}}{\mathbf{H}'_{ij}} + 2\frac{(\Omega^+ \mathbf{H}'\mathbf{D})_{ij}}{\mathbf{H}'_{ij}} + 2\frac{(\Phi^+ \mathbf{D})_{ij}\mathbf{H}'_{ij}}{\mathbf{H}^2_{ij}} + 2\frac{(\Omega^- \mathbf{H}'\mathbf{D})_{ij}\mathbf{H}'_{\mathbf{ij}}}{\mathbf{H}^2_{ij}} \\ \text{else: } 0 \end{cases} \tag{6.53}$$

Clearly, the Hessian of $\mathscr{A}(\mathbf{H}, \mathbf{H}')$ is a diagonal matrix with non-negative entries, indicating that $\mathscr{A}(\mathbf{H}, \mathbf{H}')$ is convex, as was to be shown.

Finally, to conclude the proof of Lemma 4, I show that the iterative update formula given by (6.40) is additionally optimal in the sense that it enforces non-negativity for the matrix $\mathbf{H}$. To this end, I define a matrix $\mathbf{\Lambda} \in \mathbb{R}^{k \times n}$ of Lagrangian multipliers. This gives the following associated Lagrangian:

$$\begin{aligned} F(\mathbf{H})_\Lambda = tr[\mathbf{X}\mathbf{D}\mathbf{X}^T] - 2tr[\mathbf{H}^T\mathbf{\Phi}^+\mathbf{D}] + 2tr[\mathbf{H}^T\mathbf{\Phi}^-\mathbf{D}] \\ + tr[\Omega^+\mathbf{H}\mathbf{D}\mathbf{H}^T] - tr[\Omega^-\mathbf{H}\mathbf{D}\mathbf{H}^T] - \Lambda \odot \mathbf{H} \end{aligned} \tag{6.54}$$

where $\odot$ denotes the Hadamard product. The gradient of the Lagrangian is therefore:

$$\nabla_H F(\mathbf{H})_\Lambda = -2\Phi^+\mathbf{D} + 2\Phi^-\mathbf{D} + 2\Omega^+\mathbf{H}\mathbf{D} - 2\Omega^-\mathbf{H}\mathbf{D} - \Lambda \tag{6.55}$$

where I use the identity $\nabla_H(tr[\Omega^+\mathbf{H}\mathbf{D}\mathbf{H}^T]) = 2\Omega^+\mathbf{H}\mathbf{D}$.

The Karush-Kuhn-Tucker (KKT) conditions [165] dictate that a necessary condition for optimality with the prescribed non-negative constraints is $\mathbf{H}^* \odot \mathbf{\Lambda} = 0$, where $\mathbf{H}^*$

is optimal. This indicates that an optimal solution necessarily satisfies:

$$-\Phi^{+}\mathbf{D}+\Phi^{-}\mathbf{D}+\Omega^{+}\mathbf{HD}-\Omega^{-}\mathbf{HD}-\frac{1}{2}\Lambda=0 \tag{6.56}$$

which implies the following by the KKT slackness condition:

$$\mathbf{H}_{ij}(-\Phi^{+}\mathbf{D}+\Phi^{-}\mathbf{D}+\Omega^{+}\mathbf{HD}-\Omega^{-}\mathbf{HD})_{ij}=0 \tag{6.57}$$

Equivalently, the optimal solution satisfies:

$$\mathbf{H}_{ij}^{2}(-\Phi^{+}\mathbf{D}+\Phi^{-}\mathbf{D}+\Omega^{+}\mathbf{HD}-\Omega^{-}\mathbf{HD})_{ij}=0 \tag{6.58}$$

If I solve (6.58) for $\mathbf{H}_{ij}$ I arrive at formula (6.40). This concludes the proof of Lemma 4. □

**Lemma 5.** Let $H(t)$ and $H(t+1)$ represent consecutive updates for $\mathbf{H}$ as prescribed by (6.40). Under this updating rule, the following inequality holds:

$$\begin{aligned} &tr[(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t+1))\mathbf{D}(t)(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t+1))^{T}] \\ &\leq tr[(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))^{T})] \end{aligned} \tag{6.59}$$

**Proof.** The proof of Lemma 5 follows directly from Lemma 4 and Lemma 3. □

**Lemma 6.** Under the update rule of (6.40), the following inequality holds:

$$\begin{aligned} &\|\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t+1)\|_{2,1}-\|\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t)\|_{2,1} \\ \leq \frac{1}{2}\Big[&tr[(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t+1))\mathbf{D}(t)(\mathbf{X}-\mathbf{W}(t)(\mathbf{H}(t+1))^T]- \\ &tr[(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))^T]\Big] \end{aligned} \tag{6.60}$$

**Proof.** Notice that:

$$\begin{aligned} &tr[(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))^T)] \\ =\sum_{j=1}^{p}\sum_{i=1}^{n}&(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t))_{ij}^2 D_{ii}=\sum_{i=1}^{n}\|\mathbf{x}^{(i)}-\mathbf{W}(t)\mathbf{h}(t)^{(i)}\|^2 D_{ii}(t) \end{aligned} \tag{6.61}$$

Analogously,

$$\begin{aligned} &tr[(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t+1))\mathbf{D}(t)(\mathbf{X}-\mathbf{W}(t)\mathbf{H}(t+1)^T))] \\ &=\sum_{i=1}^{n}\|\mathbf{x}^{(i)}-\mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\|^2 D_{ii}(t) \end{aligned} \tag{6.62}$$

Therefore,

$$\begin{aligned} RHS=\frac{1}{2}\sum_{i=1}^{n}&\Big(\|\mathbf{x}^{(i)}-\mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\|^2 D_{ii}(t)-\|\mathbf{x}^{(i)}-\mathbf{W}(t)\mathbf{h}(t)^{(i)}\|^2 D_{ii}(t)\Big) \\ &=\frac{1}{2}\sum_{i=1}^{n}\Big(\|\mathbf{x}^{(i)}-\mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\|^2 D_{ii}(t)-\frac{1}{D_{ii}(t)}\Big) \end{aligned} \tag{6.63}$$

Similarly,

$$\begin{aligned} LHS &= \sum_{i=1}^{n} \Big( \|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\| - \|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t)^{(i)}\| \Big) \\ &= \sum_{i=1}^{n} \Big( \|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\| - \frac{1}{D_{ii}} \Big) \end{aligned} \tag{6.64}$$

Ergo,

$$\begin{aligned} LHS - RHS &= \sum_{i=1}^{n} \Big( \|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\| - \frac{1}{2}\|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\|^2 D_{ii}(t) - \frac{1}{2}\frac{1}{D_{ii}(t)} \Big) \\ &= \sum_{i=1}^{n} \frac{D_{ii}(t)}{2} \Big( 2\frac{\|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\|}{D_{ii}(t)} - \|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\|^2 - \frac{1}{D_{ii}(t)^2} \Big) \\ &= \sum_{i=1}^{n} \frac{-D_{ii}(t)}{2} \Big( \|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\|^2 - 2\|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\| \frac{1}{D_{ii}(t)} + \frac{1}{D_{ii}(t)^2} \Big) \\ &= \sum_{i=1}^{n} \frac{-D_{ii}(t)}{2} \Big( \|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t+1)^{(i)}\| - \frac{1}{D_{ii}(t)} \Big)^2 \\ &\leq 0 \end{aligned} \tag{6.65}$$

As was to be shown. □

**Theorem 2.** Updating $\mathbf{H}$ using formula (6.40) while fixing $\mathbf{W}$ yields a monotonic decrease in the objective function defined by (6.12).

**Proof.** By Lemma 5:

$$\begin{aligned} tr[(\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t+1))\mathbf{D}(t)(\mathbf{X} - \mathbf{W}(t)(\mathbf{H}(t+1))^T] - \\ tr[(\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t))\mathbf{D}(t)(\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t))^T] \leq 0 \end{aligned} \tag{6.66}$$

Then by Lemma 6:

$$\|\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t+1)\|_{2,1} - \|\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t)\|_{2,1} \leq 0$$

That is,

$$\|\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t+1)\|_{2,1} + \overline{\alpha}\|\mathbf{W}(t)\|_2^2 \leq \|\mathbf{X} - \mathbf{W}(t)\mathbf{H}(t)\|_{2,1} + \overline{\alpha}\|\mathbf{W}(t)\|_2^2 \quad (6.67)$$

Thus proving Theorem 2. □

I now present my Regularized, L21 Semi-Nonnegative Matrix Factorization Algorithm.

**Algorithm 6.1:** Regularized L21 SNF

Initialize $\mathbf{H}(0)$ as non-negative matrix, initialize $\mathbf{W}(0)$ (e.g. use k-means)

**for** $t$ in $0 : T-1$ **do**

(1) $\mathbf{H}_{ij}(t+1) = \mathbf{H}_{ij}(t)\sqrt{\frac{(\Phi^+\mathbf{D}(t))_{ij} + (\Omega^-\mathbf{H}(t)\mathbf{D}(t))_{ij}}{(\Phi^-\mathbf{D}(t))_{ij} + (\Omega^+\mathbf{H}(t)\mathbf{D}(t))_{ij}}}$

(2)$\mathbf{W}(t+1) = [\mathbf{X}\mathbf{D}(t)\mathbf{H}(t)^T][\alpha\mathbf{I} + \mathbf{H}(t)\mathbf{D}(t)\mathbf{H}(t)^T]^{-1}$

where $\mathbf{W}^T(t)\mathbf{W}(t) = \Omega$, $\Omega = \Omega^+ - \Omega^-$, $\mathbf{W}^T(t)\mathbf{X} = \Phi$, $\Phi = \Phi^+ - \Phi^-$, and $\mathbf{D}(t)_{ii} = 1/\|\mathbf{x}^{(i)} - \mathbf{W}(t)\mathbf{h}(t)^{(i)}\|_2$.

## 6.4 Experimental Results

I perform two general experiments to compare the performance of L21 SNF Algorithm with SNF [163]: (1) general data compression via matrix factorization, and (2) qualitative facial image data reconstruction via matrix factorization. To compare general data compression performance, I begin with randomized, mixed sign data matrices

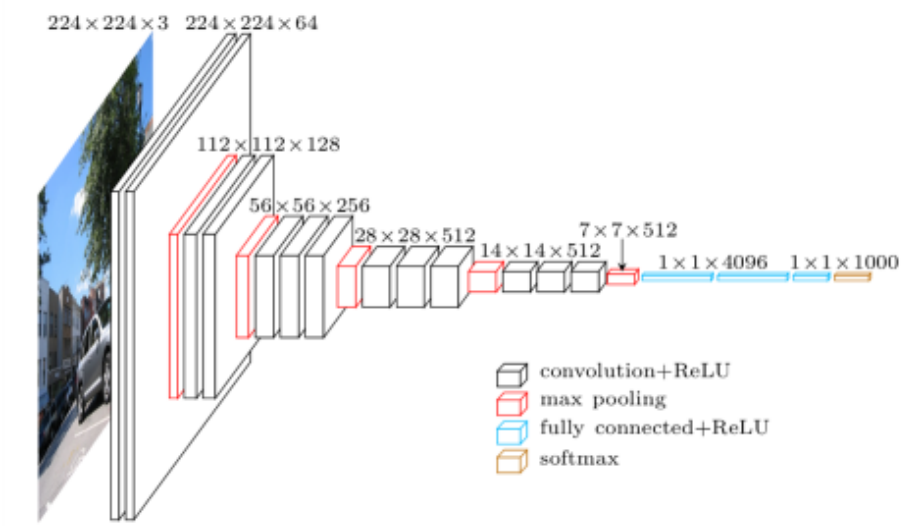


Figure 6.2: Schematic of the VGG-16 deep CNN architecture [169].

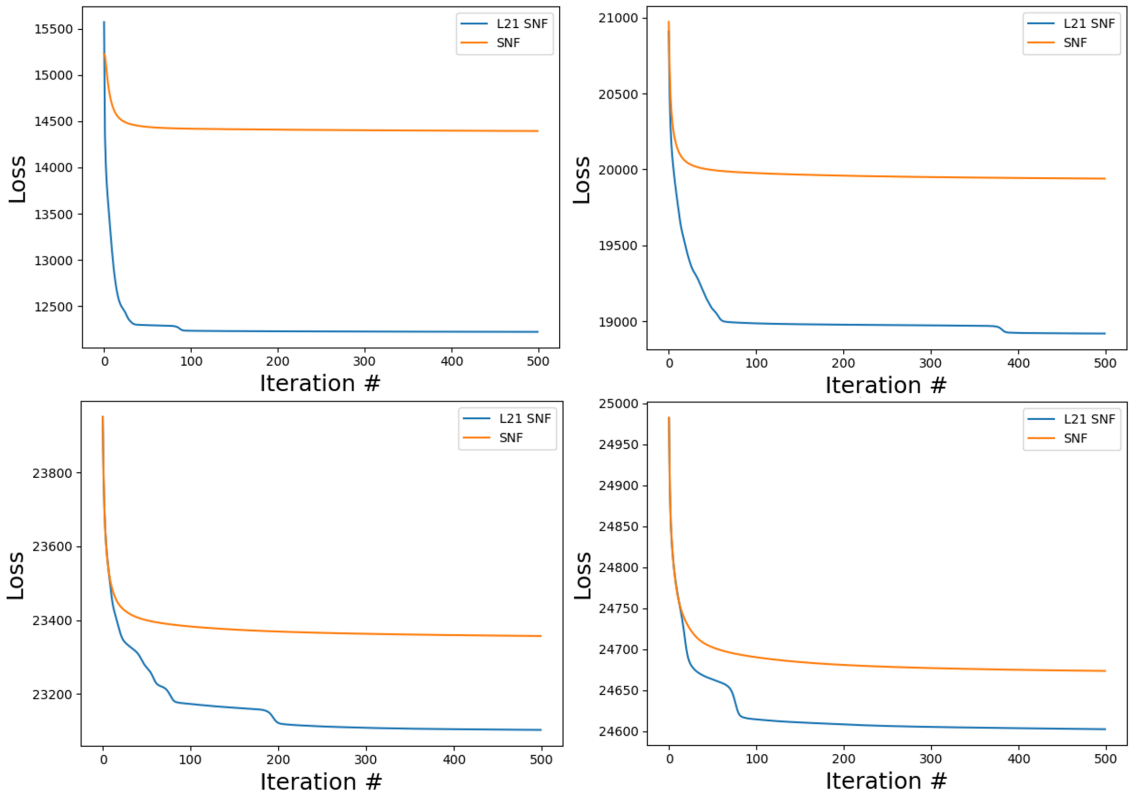


Figure 6.3: Comparison of L21 loss for L21 SNF (mine) vs SNF algorithms for compression of matrix $\mathbf{X}$ of dimension $500 \times 100$: (i) Top-Left, $500 \times 50$ compression, (ii) Top-Right, $500 \times 25$, (iii) Bottom-Left, $500 \times 10$, and (iv) Bottom-Right, $500 \times 5$.

(in the range $[-20, 20]$) $\mathbf{X}$ of dimension $500 \times 100$ and $10,000 \times 128$, respectively. In each case, I perform different degrees of compression; in the former case, through separate trials, I reduce $\mathbf{X}$ to dimension $500 \times 50$, $500 \times 25$, $500 \times 10$ and $500 \times 5$; in the latter case, I reduce $\mathbf{X}$ to dimension $10,000 \times 64$, $10,000 \times 32$, $10,000 \times 16$, and $10,000 \times 8$. In particular, these more extreme, highly non-square matrix dimensions are inspired by the use-case of deep CNN compression. For example, 'conv2' in the VGG-16 architecture [169] is of dimension $12,544 \times 128$ (when convolutional filters are flattened), see Figure 6.2.

Using identical initialization schemes for $\mathbf{W}$ and $\mathbf{H}$, I compare reconstruction loss using (4) different metrics: (i) Frobenius loss (FL), (ii) normalized Frobenius loss (NFL), i.e., $\frac{\|X-WH\|_F}{\|X\|_F}$ , (iii) L21 loss (L21), and (iv) normalized L21 loss (NL21) , i.e. $\frac{\|X-WH\|_{2,1}}{\|X\|_{2,1}}$. Table 6.1 and Table 6.2, together with Figure 6.3 and Figure 6.4, summarize these findings below. In all my of experiments, I optimize the regularization hyperparameter $\alpha$ using random search [172] over the interval [0,1]. In general, a prudent choice of $\alpha$ can improve the convergence time and overall stability of the L21 SNF algorithm in addition to reducing reconstruction loss.

Overall, the L21 SNF algorithm demonstrates a substantial improvement in comparison with SNF [163] and PCA in reducing L21-based reconstruction loss across each of my experiments, while at the same time maintaining generally strong results for L2-based reconstruction loss (see Figures 6.3 and 6.4, Tables 6.1 and 6.2). In particular, L21 SNF exhibits significant gains in the case of severely overdetermined systems. In experimental trials of reducing random, mixed sign matrices of initial dimension $10,000 \times 128$, for instance, L21 SNF shows a relative improvement of 26% over SNF for the 50% compression task, while exhibiting only a 4% increase in L2 loss, comparatively; similarly, for the 75% compression task, L21 SNF demonstrates an 11% relative improvement over SNF with respect to L21 loss, and only a 1% in-

| Compression | NFL (mine) | NL21 (mine) | NFL (SNF) | NL21 (SNF) |
|---|---|---|---|---|
| $500 \times 50$ | 0.660 (0.623) | **0.474** (**0.473**) | 0.562 (0.561) | **0.561** (**0.560**) |
| $500 \times 25$ | 0.829 (0.797) | **0.736** (**0.733**) | 0.773 (0.772) | **0.772** (**0.771**) |
| $500 \times 10$ | 0.914 (0.910) | 0.896 (0.895) | 0.907 (0.906) | 0.905 (0.904) |
| $500 \times 5$ | 0.955 ( 0.953) | 0.949 (0.949) | 0.952 (0.952) | 0.952 (0.952) |

Table 6.1: Summary of loss measures for L21 SNF algorithm (mine) vs SNF run for 500 iterations, beginning with random, mixed sign matrix of dimension $500 \times 100$. Numerical values indicate *median* value at convergence; values in parentheses indicate *minimum* values at convergence.

crease in L2 loss compared with SNF (see Table 6.2).

Lastly, I compare compression quality rendered by L21 SNF with SNF for the task of compression on a batch of images. For this experiment, I randomly sampled 200 images from the *Large-scale CelebFaces Attributes (CelebA) Dataset* [168]. Each image is of dimension $89 \times 108$; I flattened and concatenated this batch of images, rendering a data matrix $\mathbf{X}$ of dimension $9,612 \times 200$. I then ran each of the L21 SNF and SNF algorithms for 250 iterations, reducing the original matrix to size $9,612 \times 100$. The results of this experiment are shown in Figure 6.6.

Figure 6.6 in particular provides a qualitative illustration of the stark contrast in performance between L21 SNF and SNF [163] for compression applied to severely overdetermined datasets. While the reconstruction fidelity for L21 SNF is comparable with the original images from the CelebA dataset, both the SNF and PCA techniques performed poorly by comparison, as each introduces a significant amount of distortion and image artifacts in the reconstruction process.

| Compression | NFL (mine) | NL21 (mine) | NFL (SNF) | NL21 (SNF) |
|---|---|---|---|---|
| $10k \times 64$ | 0.704 (0.694) | **0.498** (**0.498**) | 0.674 (0.673) | **0.672** (**0.620**) |
| $10k \times 32$ | 0.865 (0.855) | **0.749** (**0.749**) | 0.845 (0.846) | **0.845** (**0.844**) |
| $10k \times 16$ | 0.935 (0.929) | 0.874 (0.874) | 0.925 (0.924) | 0.924 (0.923) |
| $10k \times 8$ | 0.968 (0.964) | 0.937 (0.937) | 0.962 (0.962) | 0.962 (0.962) |

Table 6.2: Summary of loss measures for L21 SNF algorithm (mine) vs SNF run for 100 iterations, beginning with random, mixed sign matrix of dimension $10,000 \times 128$.

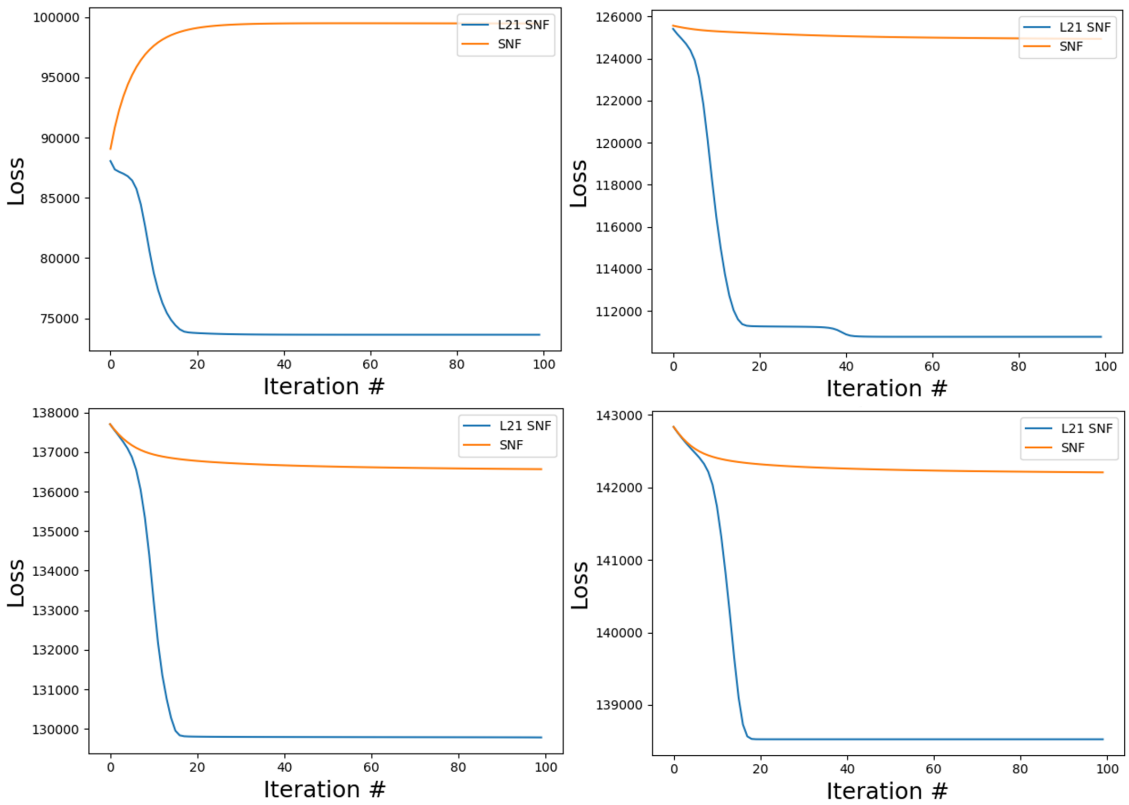


Figure 6.4: Comparison of L21 loss for L21 SNF (mine) vs SNF algorithms for compression of matrix $\mathbf{X}$ of dimension $10,000 \times 128$: (i) Top-Left, $10,000 \times 64$ compression, (ii) Top-Right, $10,000 \times 32$, (iii) Bottom-Left, $10,000 \times 16$, and (iv) Bottom-Right, $10,000 \times 8$.

Figure 6.5: Left: Original image of resolution $400 \times 400$; Middle: results using SNF to reduce image to $400 \times 50$ (500 iterations); Right: results using L21 SNF (mine) to reduce image to $400 \times 50$ (500 iterations). Notice that even though L21 SNF is optimized for L21 loss, the two compression results exhibit nearly identical reconstruction fidelity, which is to say that L21 SNF also maintains strong results with respect to Frobenius loss.

Figure 6.6: Results for compression of batch of 200 face images sampled from the CelebA [168] dataset; I show a sample of seven randomly selected images after compression. The original image batch of dimension $9,612 \times 200$ was compressed to $9,612 \times 100$; each algorithm was run for 250 iterations. Top: ground-truth images; Second from Top: L21 SNF (mine) rendered result; Second from Bottom: SNF results; Bottom: PCA results.

# Chapter 7

# Conclusions

This work has provided several innovations that accentuate and unify non-parametric and deep learning methodologies across a diverse array of computer vision processes. I illustrate in particular that these paradigms can be successfully leveraged to provide a basis for higher order analytical modalities, including "visual situation recognition" necessary for the safe and effective deployment of deep models in real-world, risk-sensitive settings.

In Chapter 1, I introduced the general problem scope of my dissertation. In Chapters 2 and 3, I demonstrated that prior knowledge and situation-relevant context can be encoded with a dynamic, non-parametric "situation model." Moreover, this model, when used in conjunction with deep learning, can accurately and efficiently localize relevant objects in an image. In particular, I introduce the MIC-Situate algorithm which presents the notion of a context-based importance cluster for rendering situational-specific conditional density estimates; in conjunction with this technique, I present a multipole expansion with stochastic filtering-based method to reduce KDE approximations employed in active object search from $O(MN)$ complexity to $O(M + N)$.

In Chapter 4, I extended these ideas through the development of a more statistically-principled approach to object localization. Here I presented a novel technique for the challenging task of efficient object localization. My method trains a predicted-offset

model, demonstrating successfully the ability of CNN-based features to serve as the input for an object localization method. Using Bayesian optimization, this technique surpasses the state-of-the-art regression method employed in R-CNN (and its extensions) for the localization of pedestrians in high-resolution still images with computational efficiency. With future research, this work can potentially be extended to gradient-based GPs and massively scalable GPs, whereby GP-enabled localization can directly incorporate bounding-box size parameters, as well as leverage additional sources of visual context for localization. More generally, I aim to apply these approaches to broader, high-dimensional problem regimes.

In Chapter 5, I applied a Bayesian optimization approach to the more extreme, "one-shot" use-case of video tracking. In this work I presented the first integrated dynamic Bayesian optimization framework in combination with deep learning for video tracking. While this algorithm demonstrated effectiveness in video tracking, I believe it can be further improved in the future. I intend to potentially expand the current approach to accommodate the following enhancements: (1) GP-enabled multi-scaling (so the GP is generated in five dimensions, including space, size and time); (2) adaptive Bayesian optimization (ABO) which adaptively alters the bounds and sample constraints at each frame for optimizing the acquisition function based on the learned time-related length-scale parameter; (3) I anticipate furthermore that incorporating a fully-convolutional architecture into the Siamense conv-net with my current pipeline will yield faster than real-time video tracking with the added benefits of Bayesian non-parametrics. This research can be further augmented to include visual context models for structured image and video types to be used with video scene and behavior recognition; various numerical optimization techniques can further improve the efficiency and speed of the GP-based video tracking. I believe that this research has significant potential for widespread real-world use, including applications

to surveillance, high-level scene understanding in computer vision systems, and a myriad of commercial and consumer-based applications.

Finally, in Chapter 6, I devised a novel algorithm, "Regularized L21-Based Semi-NonNegative Matrix Factorization" which provides a general, additive, parts-based data compression algorithm, using an L21-norm-based loss function. In particular, I demonstrated a formal proof of convergence of this algorithm. Through experiments, I showed the use-case advantages presented by my algorithm in comparison with baseline compression algorithms, including SNF and PCA. In particular, my Regularized L21-Based Semi-NonNegative Matrix Factorization algorithm is particularly well-suited to highly overdetermined systems. In future work, I plan to extend this algorithm to potentially incorporate sparsity constraints, and to furthermore deploy it for real-world applications, including: deep model compression and genomic data compression.

## References


[1]"Portland State Dog-Walking Images." [Online]. Available: http://www.cs.pdx.edu/mm/PortlandStateDogWalkingImages.html

[2] D. R. Hofstadter, Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought. New York: Basic Books, 1995.

[3] ——, "Analogy as the core of cognition," in The Analogical Mind: Perspectives from Cognitive Science, D. Gentner, K. J. Holyoak, and B. N. Kokinov, Eds. MIT Press, 2001, pp. 499–538.

[4] "Dog Swimming." [Online]. Available: http://www.drollnation.com/gallery/2015/12/randomness-120315-5.jpg

[5] "Dog Walking Dog." [Online]. Available: http://www.delcopetcare.com/wp-content/uploads/2013/02/dogwalking.jpg

[6] D. Hofstadter and E. Sander, Surfaces and Essences. Basic Books, 2013.

[7] D. R. Hofstadter and M. Mitchell, "The Copycat project: A model of mental fluidity and analogy-making," in Advances in Connectionist and Neural Computation Theory, K. Holyoak and J. Barnden, Eds. Ablex Publishing Corporation, 1994, vol. 2, pp. 31–112.

[8] M. Mitchell, Analogy-Making as Perception: A Computer Model. MIT Press, 1993.

[9] G. Guo and A. Lai, "A survey on still image based human action recognition," Pattern Recognition, vol. 47, no. 10, pp. 3343–3361, 2014.

[10] S. Gupta and J. Malik, "Visual Semantic Role Labeling," arXiv:1505.04474, 2015.

[11] L.-J. Li and L. Li Fei-Fei, "What, where and who? Classifying events by scene and object recognition," in International Conference on Computer Vision (ICCV). IEEE, 2007, pp. 1–8.

[12] L. Wang, Z. Zhe Wang, W. Wenbin Du, and Y. Qiao, "Object-scene convolutional neural networks for event recognition in images," in Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. IEEE, 2015, pp. 30–35.

[13] A. Yatskar, M, Zettlemoyer, L., Farhadi, "Situation recognition: Visual semantic role labeling for image understanding," in Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.

[14] M. Bar, "Visual objects in context." Nature Reviews Neuroscience, vol. 5, no. 8, pp. 617–629, 2004.

[15] P. G. Malcolm, George L. and Schyns, "More than meets the eye: The active selection of diagnostic information across spatial locations and scales during scene categorization," in Scene Vision: Making Sense of What We See, M. Kveraga, K. and Bar, Ed. MIT Press, 2014, pp. 27–44.

[16] M. Neider and G. Zelinsky, "Scene context guides eye movements during visual search," Vision Research, vol. 46, no. 5, pp. 614–621, 2006.

[17] M. C. Potter, "Meaning in visual search," Science, vol. 187, pp. 965–966, 1975.

[18] C. Summerfield and T. Egner, "Expectation (and attention) in visual cognition." Trends in Cognitive Sciences, vol. 13, no. 9, pp. 403–9, 2009.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv:1512.03385, 2015.

[20] R. Girshick, "Fast R-CNN," in International Conference on Computer Vision (ICCV). IEEE, 2015, pp. 1440–1448.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Uni-

fied, real-time object detection," arXiv:1506.02640, 2015.

[22] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338, 2010.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.

[24] G. C. H. E. de Croon, E. O. Postma, and H. J. van den Herik, "Adaptive gaze control for object detection." Cognitive Computation, vol. 3, no. 1, pp. 264–278, 2011.

[25] A. Gonzalez-Garcia, A. Vezhnevets, and V. Ferrari, "An active search strategy for efficient object class detection," in Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015, pp. 3022–3031.

[26] Y. Lu, T. Javidi, and S. Lazebnik, "Adaptive object detection using adjacency and zoom prediction," arXiv:1512.07711, 2015.

[27] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in International Conference on Computer Vision (ICCV). IEEE, 2011, pp. 1331–1338.

[28] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in International Conference on Computer Vision (ICCV). IEEE, 2015, pp. 2488–2496.

[29] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 11, pp. 1254–1259, 1998.

[30] L. Elazary and L. Itti, "A Bayesian model for efficient visual search and recogni-

tion," Vision Research, vol. 50, no. 14, pp. 1338–1352, 2010.

[31] L. Itti and C. Koch, "Computational modelling of visual attention," Nature Reviews Neuroscience, vol. 2, no. 3, pp. 194–203, 2001.

[32] S. Frintrop, T. Werner, and G. M. Garcia, "Traditional saliency reloaded: A good old model in new shape," in Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015, pp. 82–90.

[33] S. Manen, M. Guillaumin, and L. V. Gool, "Prime object proposals with randomized Prim's algorithm," in International Conference on Computer Vision (ICCV). IEEE, 2013, pp. 2536–2543.

[34] J. Hosang, R. Benenson, and B. Schiele, "How good are detection proposals, really?" arXiv:1406.6962, 2014.

[35] J. Oramas-M. and T. Tuytelaars, "Recovering hard-to-find object instances by sampling context-based object proposals," arXiv:1511.01954, 2015.

[36] A. Rosenfeld and S. Ullman, "Visual Concept Recognition and Localization via Iterative Introspection," arXiv:1603.04186, 2016.

[37] S. Singh, D. Hoiem, and D. Forsyth, "Learning to Localize Little Landmarks," in Conference on Computer Vision and Pattern Recognition. IEEE, 2016.

[38] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," in Conference on Computer Vision and Pattern Recognition. IEEE, 2010, pp. 17–24.

[39] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627–45, 2010.

[40] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2014, pp. 580–587.

[41] C. Galleguillos and S. Belongie, "Context based object categorization: A critical survey," Computer Vision and Image Understanding, vol. 114, no. 6, pp. 712–722, 2010.

[42] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2010, pp. 129–136.

[43] X. Cao, X. Wei, Y. Han, and X. Chen, "An object-level high-order contextual descriptor based on semantic, spatial, and scale cues." IEEE Transactions on Cybernetics, vol. 45, no. 7, pp. 1327–39, 2015.

[44] S. Divvala and D. Hoiem, "An empirical study of context in object detection," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1271 – 1278, 2009.

[45] C. Galleguillos, "Object categorization using co-occurrence, location and appearance," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[46] S. Marat and L. Itti, "Influence of the amount of context learned for improving object classification when simultaneously learning object and contextual cues," Visual Cognition, vol. 20, no. 4-5, pp. 580–602, 2012.

[47] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2014, pp. 891–898.

[48] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov, "Scalable, high-quality object detection," arXiv:1412.1441, 2014.

[49] A. Torralba and K. Murphy, "Context-based vision system for place and object recognition," in Ninth IEEE International Conference on Computer Vision (ICCV). IEEE, 2003, pp. 273–280.

[50] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," International Journal of Computer Vision, vol. 80, no. 3, pp. 300–316, 2008.

[51] Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler, "SegDeepM: Exploiting segmentation and context in deep neural networks for object detection," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 07-12-June, 2015, pp. 4703–4711.

[52] G. Heitz and D. Koller, "Learning spatial context: Using stuff to find things," in European Conference on Computer Vision (ECCV), 2008, pp. 30–43.

[53] H. Izadinia, F. Sadeghi, and A. Farhadi, "Incorporating scene context and object layout into appearance modeling," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2014, pp. 232–239.

[54] W. Chu and D. Cai, "Deep feature based contextual model for object detection," arxiv:1604.04048, apr 2016.

[55] D. H. Ballard, "Animate vision," Artificial Intelligence, vol. 48, no. 1, pp. 57–86, 1991.

[56] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 1, pp. 185–207, 2013.

[57] B. Alexe, N. Heess, Y. Teh, and V. Ferrari, "Searching for objects driven by context," in Advances in Neural Information Processing Systems, 2012, pp. 1–9.

[58] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in Advances in Neural Information Processing Systems (NIPS), 2014, pp. 2204–2212.

[59] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," arXiv:1412.7755, 2014.

[60] X. S. Chen, H. He, and L. S. Davis, "Object detection in 20 questions," in 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2016, pp. 1–9.

[61] V. K. Nagaraja, V. I. Morariu, and L. S. Davis, "Searching for Objects using Structure in Indoor Scenes," arXiv:1511.07710, 2015.

[62] D. Hofstadter and E. Sander, Surfaces and Essences. Basic Books, 2013.

[63] M. Bar, "Visual objects in context," Nature Reviews Neuroscience, vol. 5, no. 8, pp. 617–629, 2004.

[64] G. L. Malcolm and P. G. Schyns, "More than meets the eye: The active selection of diagnostic information across spatial locations and scales during scene categorization," in Scene Vision: Making Sense of What We See, K. Kveraga and M. Bar, Eds. MIT Press, 2014, pp. 27–44.

[65] M. Neider and G. Zelinsky, "Scene context guides eye movements during visual search," Vision Research, vol. 46, no. 5, pp. 614–621, 2006.

[66] M. C. Potter, "Meaning in visual search," Science, vol. 187, pp. 965–966, 1975.

[67] C. Summerfield and T. Egner, "Expectation (and attention) in visual cognition." Trends in Cognitive Sciences, vol. 13, no. 9, pp. 403–9, 2009.

[68] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338, 2010.

[69] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv:1512.03385, 2015.

[70] R. Girshick, "Fast R-CNN," in International Conference on Computer Vision (ICCV). IEEE, 2015, pp. 1440–1448.

[71] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," arXiv:1506.02640, 2015.

[72] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.

[73] G. C. H. E. de Croon, E. O. Postma, and H. J. van den Herik, "Adaptive gaze control for object detection." Cognitive Computation, vol. 3, no. 1, pp. 264–278, 2011.

[74] A. Gonzalez-Garcia, A. Vezhnevets, and V. Ferrari, "An active search strategy for efficient object class detection," in Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015, pp. 3022–3031.

[75] Y. Lu, T. Javidi, and S. Lazebnik, "Adaptive object detection using adjacency and zoom prediction," arXiv:1512.07711, 2015.

[76] M. H. Quinn, A. D. Rhodes, and M. Mitchell, "Active object localization in visual situations," arXiv:1607.00548, 2016.

[77] S. Manen, M. Guillaumin, and L. V. Gool, "Prime object proposals with randomized Prim's algorithm," in International Conference on Computer Vision (ICCV). IEEE, 2013, pp. 2536–2543.

[78] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in 2010 IEEE International Conference on Data Mining. IEEE, 2010, pp. 911–916.

[79] C. G. Lambert, S. E. Harrington, C. R. Harvey, and A. Glodjo, "Efficient on-line nonparametric kernel density estimation," Algorithmica, vol. 25, no. 1, pp. 37–57, 1999.

[80] C. Yang, R. Duraiswami, and L. S. Davis, "Efficient kernel machines using the improved fast Gauss transform," in Advances in neural information processing systems, 2004, pp. 1561–1568.

[81] G. Fasshauer, "Positive definite kernels: past, present and future," Dolomite Research Notes on Approximation, 2011.

[82] M. G. Genton, "Classes of Kernels for Machine Learning: A Statistics Perspective," Journal of Machine Learning Research, vol. 2, no. Dec, pp. 299–312, 2001.

[83] M. Sugiyama, I. Takeuchi, T. Suzuki, and T. Kanamori, "Conditional Density Estimation via Least-Squares Density Ratio Estimation," AISTATS, pp. 781–788, 2010.

[84] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," Theoretical Computer Science, vol. 38, pp. 293–306, 1985.

[85] T. Feder and D. Greene, "Optimal algorithms for approximate clustering," in Proceedings of the twentieth annual ACM symposium on Theory of computing - STOC '88. New York, New York, USA: ACM Press, 1988, pp. 434–444.

[86] Q. Liu, D. Pitt, X. Zhang, and X. Wu, "A Bayesian Approach to Parameter Estimation for Kernel Density Estimation via Transformations," Annals of Actuarial Science, vol. 5, no. 2, pp. 181–193, 2011.

[87] M. Mitchell, Analogy-Making as Perception: A Computer Model. MIT Press, 1993.

[88] Roberto Brunelli. 2009. Template Matching Techniques in Computer Vision: Theory and Practice. Wiley Publishing.

[89] David G. Lowe. 1999. Object Recognition from Local Scale-Invariant Features. In Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2 (ICCV '99), Vol. 2. IEEE Computer Society, Washington, DC, USA, 1150-.

[90] George Nebehay and Roman P. Pflugfelder. 2014. Consensusbased Matching and Tracking of Keypoints for Object Tracking. In (IEEE) Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, 862-869.

[91] Avneet Dalal and Bill Triggs. 2005. Histograms of Oriented Gradients for Human

Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01 (CVPR '05), Vol. 1. IEEE Computer Society, Washington, DC, USA, 886-893.

[92] George Nebehay and Roman P. Pflugfelder. 2015. Clustering of Static-Adaptive Correspondences for Deformable Object Tracking. In (IEEE) Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2784-2791.

[93] Dorin Comaniciu and Peter Meer. 2002. Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24, 5 (May 2002), 603-619. DOI=http://dx.doi.org/10.1109/34.1000236

[94] B. A. Draper, D. S. Bolme, J. R. Beveridge and Y. M. Lui, "Visual object tracking using adaptive correlation filters," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR), San Francisco, CA, USA, 2010, pp. 2544-2550. doi:10.1109/CVPR.2010.5539960

[95] "MUlti-Store Tracker (MUSTer): a Cognitive Psychology Inspired Approach to Object Tracking", Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015, Boston, USA

[96] Ross Girshick. 2015. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV '15). IEEE Computer Society, Washington, DC, USA, 1440-1448.
DOI=http://dx.doi.org/10.1109/ICCV.2015.169

[97] David Held and Sebastian Thrun and Silvio Savarese. 2017. Learning to Track at 100 FPS with Deep Regression Networks. European Conference on Computer Vision (ECCV). Springer

[98] Zhenhua Fan, Hongbing Ji, and Yongquan Zhang. 2015. Iterative particle filter for visual tracking. Image Commun. 36, C (August 2015), 140-153.

[99] Shiuh-Ku Weng, Chung-Ming Kuo, and Shu-Kang Tu. 2006. Video object tracking using adaptive Kalman filter. J. Vis. Comun. Image Represent. 17, 6 (December 2006), 1190-1208.

[100] R. Tao and E. Gavves and A. Smeulders. 2016. Siamese Instance Search for Tracking. Computer Vision and Pattern Recognition (CVPR).

[102] Koch, Gregory, Zemel, Richard and Salakhutdinov, Ruslan. "Siamese Neural Networks for One-shot Image Recognition." Paper presented at the meeting of the , 2015.

[103] Bertinetto, Luca et al. Fully-Convolutional Siamese Networks for Object Tracking. Computer Vision ECCV 2016 Workshops (2016): 850865. Crossref. Web.

[104] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12), F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Vol. 1. Curran Associates Inc., USA, 1097-1105.

[105] Yann LeCun and Yoshua Bengio. 1998. Convolutional networks for images, speech, and time series. In The handbook of brain theory and neural networks, Michael A. Arbib (Ed.). MIT 4 Press, Cambridge, MA, USA 255-258.

[106] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.

[107] Luca Bertinetto and Jack Valmadre and Joao Henriques and Andrea Vedaldi and Phillip Torr. 2016. Fully-Convolutional Siamese Networks for Object Tracking. ECCV.

[108] Jurgen Branke. 2001. Evolutionary Optimization in Dynamic Environments.

Kluwer Academic Publishers, Norwell, MA, USA.

[109] Carl Edward Rasmussen and Christopher K. I. Williams. 2005. Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press.

[110] Tony Jebara, Risi Kondor, and Andrew Howard. 2004. Probability Product Kernels. J. Mach. Learn. Res. 5 (December 2004), 819-844.

[111] Naiyan Wang, Jianping Shi, Dit-Yan Yeung, and Jiaya Jia. 2015. Understanding and Diagnosing Visual Tracking Systems. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV '15). IEEE Computer Society, Washington, DC, USA, 3101-3109.
DOI=http://dx.doi.org/10.1109/ICCV.2015.355

[112] Matej Kristan and Jiri Matas and Alevs Leonardis and Tomas Vjori and Roman Pflugfelder and Gustavo Fernandez and George Nebehay and Fatih Porikli and Luka Vcehovin. 2016. A Novel Performance Evaluation Methodology for Single-Target Trackers. PAMI.

[113] Favour M. Nyikosa and Michael A. Osbore and Stephen J. Roberts. 2018. Bayesian Optimization for Dynamic PRoblems. arXiv.1803.03432

[114] Anthony Rhodes and Jordan Witte and Bruno Jedynak and Melanie Mitchell. 2018. Gaussian Processes with Context-Supported Priors for Active Object Localization. International Joint Conference on Neural Networks (IJCNN).

[115] Seth Flaxman and Andrew Wilson and Daniel Neill and Hannes Nickisch and Alex Smola. 2015. Fast Kronecker Inference in Gaussian Processes with non-Gaussian Likelihoods. Proceedings of the 32nd International Conference on Machine Learning.

[116] Sangdoo Yun and Jongwon Choi and Youngjoon Yoo and Kimin Yun and Jin Young Choi. Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning. The IEEE Conference on Computer Vision and Pattern Recognition

(CVPR 2017).

[117] Briechle, Kai and Hanebeck, Uwe. (2001). Template matching using fast normalized cross correlation. Proceedings of SPIE - The International Society for Optical Engineering. 4387. 10.1117/12.421129.

[118] Eric Brochu and Vlad M. Cora and Nando de Freitas. 2010. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. arXiv:1012.2599v1

[119] D. Lizotte. Practical Bayesian Optimization. PhD thesis, University of Alberta, Edmonton, Alberta, Canada, 2008.

[120] Kristan, Matej et al. 2015. The Visual Object Tracking VOT2014 Challenge Results. ECCV 2014 Workshops.

[121] Matej Kristan et al. 2017. The Visual Object Tracking VOT2016 Challenge Results. ECCV 2017 Workshops.

[122] Kervadec et al, Boundary Loss for Highly Unbalanced Segmentation. MIDL 2019 submissions.

[123] Hariharan et al, Object Instance Segmentation and Fine-Grained Localization Using Hypercolumns. PAMI 2016.

[124] Li et al, Interactive Image Segmentation with Latent Diversity. CVPR 2018.

[125] Khoreva et al, Learning Video Object Segmentation from Static Images. CVPR 2017.

[126] Kim et al. Compression of Deep Convolutional Neural Networks for Fast and Low Power Mobile Applications. ICLR 2016

[127] Huber. Robust Estimation of a Location Parameter. Annals of Mathematical Statistics 1964.

[128] Rabanser. Introduction to Tensor Decompositions and their Applications in Machine Learning. https://arxiv.org/pdf/1711.10781.pdf 2018

[129] Lathauwer et al. A Multilinear Singular Value Decomposition. SIAM Journal on Matrix Analysis and Applications 2000.

[130] Horn and Schunk. Determining Optical Flow. Artificial Intelligence, 1981.

[131] Lucas and Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. Proceedings of the Imaging Understanding Workshop.

[132] Felzenswalb and Huttenlocher. Distance Transforms of Samples Functions. Theory of Computing, 2012.

[133] Achanta, et al. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. PAMI, 2012.

[134] Bertinetto, et al. Fully-Convolutional Siamese Networks for Object Tracking. CVPR, 2017.

[135] Baltrusaitis, et al. OpenFace: An Open Source Facial Behavior Analysis Toolkit. WACV, 2016.

[136] Rother, et al. "Grabcut" – Interactive Foreground Extraction using Iterated Graph Cuts. SIGGRAPH, 2004.

[137] Tsai, et al. Video Segmentation via Object Flow. CVPR, 2016.

[138] Kohli, et al. Robust Higher Order Potential for Enforcing Label Consistency. International Journal on Computer Vision, 2009.

[139] Alex Kendall, Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? https://arxiv.org/abs/1703.04977 2018

[140] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR, 2015.

[141] Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. ICLR, 2016.

[142] Hariharan, B. et al. Hypercolumns for Object Segmentation and Fine-grained Localization. CVPR, 2015.

[143] Michael Levandowsky and David Winter. Distance Between Sets. Nature, November, 1971.

[144] He et al. Deep Residual Learning for Image Recognition. CVPR, 2016.

[145] Szegedy et al. Going deeper with convolutions. CVPR, 2015.

[146] Babajide O. Ayinde and Jacek M. Zurada. Building Efficient ConvNets Using Redundant Feature Pruning. ICLR Workshop 2018.

[147] Cheng et al. A Survey of Model Compression and Acceleration for Deep Neural Networks. IEEE Signal Processing Magazine, January, 2018.

[148] Nitish Srivastava et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. JMLR, 2014.

[149] Han et al. DSD: Dense-Sparse-Dense Training for Deep Neural Networks. ICLR, 2017.

[150] Han et al. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. ICLR, 2016.

[151] Denton et al. Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation. NIPS, 2014.

[152] V. Lebedev and V. S. Lempitsky. Fast ConvNets Using Group-Wise Brain Damage. CVPR, 2016.

[153] Nicolas Gillis. The Why and How of Non-Negative Matrix Factorization [Book Chapter]. CRC Press, 2015.

[154] Kullback, S. and Leibler, R.A. On information and sufficiency. In Annals of Mathematical Statistics. 22 (1): 79–86, 1951.

[155] Daniel D. Lee and H. Sebastian Seung. Algorithms for Non-negative Matrix Factorization NIPS, 2000.

[156] Donghui Chen and Robert J. Plemmons. Nonnegativity constraints in numerical analysis. Symposium on the Birth of Numerical Analysis, 2009.

[157] Vu et al. Combining Non-Negative Matrix Factorization and Deep Neural Networks for Speech Enhancement and Automatic Speech Recognition. ICASSP, 2016.

[158] Chih-Jen Lin. Projected Gradient Methods for Nonnegative Matrix Factorization. Neural Computation. 19 (10): 2756–2779. 2007.

[159] Amnon Shashua and Tamir Hazan. Non-Negative Tensor Factorization with Applications to Statistics and Computer Vision. ICML, 2005.

[160] Nie et al. Efficient and Robust Feature Selection via Joint 2,1-Norms Minimization. NIPS, 2010.

[161] Ding et al. R1-PCA: Rotational Invariant L1-norm Principal Component Analysis for Robust Subspace Factorization. ICML, 2006.

[162] Tamara G. Kolda and Brett W. Bader. SIAM Rev., 51(3), 455–500, 2009.

[163] C Ding, T Li, MI Jordan, Convex and semi-nonnegative matrix factorizations, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32, 45-55, 2010.

[164] Nie, Feiping and Huang, Heng and Cai, Xiao and Ding, Chris. (2010). Efficient and Robust Feature Selection via Joint L2,1-Norms Minimization.. NIPS. 1813-1821.

[165] Ruszczyński, Andrzej (2006). Nonlinear Optimization. Princeton, NJ: Princeton University Press. ISBN 978-0691119151.

[166] Petersen, K. B. and Pedersen, M. S. The Matrix Cookbook. , Technical University of Denmark (2008).

[167] Kong, Deguang, Chris H. Q. Ding and Heng Huang. "Robust nonnegative matrix factorization using L21-norm." CIKM '11 (2011).

[168] Liu, Ziwei and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou. (2015) Deep Learning Face Attributes in the Wild. Proceedings of International Conference on Computer Vision (ICCV).

[169] K. Simonyan, A. Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition International Conference on Learning Representations

(ICLR).

[170] Cheng, Y., Wang, D., Zhou, P., Zhang, T. (2017). A Survey of Model Compression and Acceleration for Deep Neural Networks. ArXiv, abs/1710.09282.

[171] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, Song Han: AMC: AutoML for Model Compression and Acceleration on Mobile Devices. ECCV (7) 2018: 815-832

[172] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13, null (February 2012), 281–305.

[173] Hernaez, Mikel et al. 2019. Genomic Data Compressio. Annual Review of Biomedical Data Science July 2019 2(1):19

[174] S. Zhang, R. Benenson, M. Omran, J. Hosang, B. Schiele, How Far are We from Solving Pedestrian Detection?, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016: pp. 1259–1267.

[175] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, in: 2012 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2012: pp. 3354–3361

[176] Xu Li, S.C. Hagness, A Confocal Microwave Imaging Algorithm for Breast Cancer Detection, in Proc. IEEE Microwave and Wireless Components Letters (Volume: 11, Issue 3, March 2001) : pp.130-132.

[177] Vicente Ordonez, Girish Kulkarni, Tamara L. Berg. Im2Text: Describing Images Using 1 Million Captioned Photographs. Neural Information Processing Systems (NIPS), 2011.

[178] P. Espinace, T. Kollar, A. Soto, N. Roy, Indoor Scene Recognition Through Object Detection, in: 2010 IEEE Conference on Robotics and Automation (ICRA).

[179] A. Morales, T. Asfour, P. Azad, S. Knoop, R. Dillmann, Integrated Grasp Planning and Visual Object Localization For a Humanoid Robot with Five-Fingered

Hands, in: 2006 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, 2006: pp. 5663–5668.

[180] J. Ren, S., He, K., Girshick, R., Sun, Faster R-CNN: Towards real-time object detection with region proposals, in: Adv. Neural Inf. Process. Syst. 28 (NIPS 2015), 2015.

[181] S. Manen, M. Guillaumin, and L. V. Gool, "Prime object proposals with randomized Prim's algorithm," in International Conference on Computer Vision (ICCV). IEEE, 2013, pp. 2536–2543.

[182] D. H. Ballard, "Animate vision," Artificial Intelligence, vol. 48, no. 1, pp. 57–86, 1991. [183] B. Alexe, N. Heess, Y. Teh, and V. Ferrari, "Searching for objects driven by context," in Advances in Neural Information Processing Systems, 2012, pp. 1–9.

[184] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 1, pp. 185–207, 2013.

[185] A. Gonzalez-Garcia, A. Vezhnevets, and V. Ferrari, "An active search strategy for efficient object class detection," in Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015, pp. 3022–3031.

[186] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in Advances in Neural Information Processing Systems (NIPS), 2014, pp. 2204–2212.

[187] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in International Conference on Computer Vision (ICCV). IEEE, 2015, pp. 2488–2496.

[188] X. S. Chen, H. He, and L. S. Davis, "Object detection in 20 questions," in 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2016,

pp. 1–9.

[189] V. K. Nagaraja, V. I. Morariu, and L. S. Davis, "Searching for Objects using Structure in Indoor Scenes," arXiv:1511.07710, 2015.